October/November 1983

An independent magazine published by ECC Publications

SINCLAIR PROJECTS

95p

THE MAGAZINE FOR THE SERIOUS SINCLAIR USER

WEATHER STATION—KEEP AN AUTOMATIC RECORD OF THE WEATHER

SAVEING AID
DESIGN GUIDE
BURGLAR ALARM
JOYSTICK ADAPTOR
BATTERY-BACKED RAM

SIX PROJECTS FOR THE SPECTRUM AND ZX-81

SINCLAIR PROJECTS

FROM THE EDITOR

IN THIS the sixth issue of *Sinclair Projects* we have again been able to bring you six interesting projects which will enhance your Spectrum or ZX-81.

Our main article which is featured on the cover is the first in a series which will allow you to build a complete weather station. Graham Bradley describes how to measure wind speed and in following issues we will be including modules for measuring light levels, rainfall, humidity pressure and temperature. The finished project will be of interest to owners of all Sinclair users as they will work with both the Spectrum and the ZX-81.

Graham has contributed to another interesting article on making a battery-backed RAM board.

Another of our writers who has two articles printed is Corin Howitt. He followed the first part of his project to build a burglar alarm with two small extensions of the basic system. The first show how to develop software to monitor the input lines and increase the flexibility of the system allowing things such as a delay on leaving and entering the house. The other is a hardware design to improve the back-up alarm monitoring when the alarm is off or otherwise engaged.

He has also written an article showing how to overcome the problem of joysticks not working with all games in which they could be used. One solution by manufacturers has been to have a compatible range of software for their products. Howitt shows how to specify the keys to be mimicked by the joystick.

LOADing programs into the ZX-81 has long been a problem for new and old users alike. Many solutions have been proposed and many little pieces of hardware have been made to try and overcome the problem. Charles Rowbotham looks at the difficulties from a different viewpoint and suggests that the answer lies in the SAVEing side of the process.

As he says most of the problems arise when trying to LOAD software SAVEd by other people then it could be that when the SAVEing is being carried out a function of the tape recorder is responsible for ensuring that programs cannot be stored. The automatic gain control of most cassette recorders means that the signal cannot be understood by the ZX-81.

The last project is a software system which enables people to simulate logic devices and for a given circuit design to print out the waveform or table of results. It is written by Malcolm Farnsworth who says it can be used for initial design, analysis or fault finding.

Editor Nigel Clark **Consultant editor** David Buckley **Production editor** Harold Mayes MBE **News writer** Stephen Adams **Design** Elaine Bishop **Advertisement manager** John Ross **Advertisement Executive** Annette Burrows **Editorial assistant** Margaret Hawkins **Managing director** Terry Cartwright **Chairman** Richard Hease.

Sinclair Projects is published bi-monthly by ECC Publications Ltd. It is in no way connected with Sinclair Research Ltd.

Telephone, all departments: 01-359 3525. If you would like to contribute to any of the *Sinclair User* group of publications please send programs, articles or ideas for hardware projects to Sinclair User and Projects, ECC Publications, 196-200 Balls Pond Road, London N1 4AQ. We pay £50 per 1,000 words for each article used.

ISSN 0264/0449. Printed and typeset by Cradley Print PLC, Warley, W. Midlands. Distributed by Spotlight Magazine Distribution Ltd, 1 Benwell Road, Holloway, London N7. 01-607 6411.

MAIL ORDER ADVERTISING
British Code of Advertising Practice

Advertisements in this publication are required to conform to the British Code of Advertising Practice. In respect of mail order advertisements where money is paid in advance, the code requires advertisers to fulfil orders within 28 days, unless a longer delivery period is stated. Where goods are returned undamaged within seven days, the purchaser's money must be refunded. Please retain proof of postage/despatch, as this may be needed.

Mail Order Protection Scheme

If you order goods from Mail Order advertisements in this magazine and pay by post in advance of delivery, *Sinclair User* will consider you for compensation if the Advertiser should become insolvent or bankrupt, provided:

(1) You have not received the goods or had your money returned; and

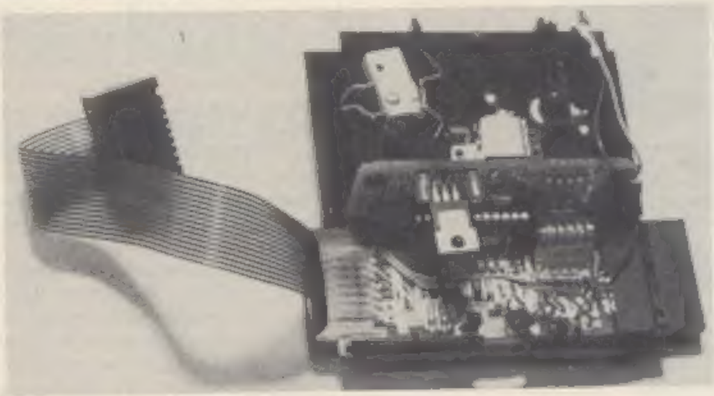
(2) You write to the Publisher of *Sinclair User* summarizing the situation not earlier than 28 days from the day you sent your order and not later than two months from that day.

Please do not wait until the last moment to inform us. When you do write, we will tell you how to make your claim and what evidence of payment is required.

We guarantee to meet claims from readers in accordance with the above procedure as soon as possible after the Advertiser has been declared bankrupt or insolvent (up to a limit of £4,250 per annum for any one Advertiser so affected and up to £10,000 per annum in respect of all insolvent Advertisers. Claims may be paid for higher amounts, or when the above procedure has not been complied with at the discretion of Sinclair User, but we do not guarantee to do so in view of the need to set some limit to this commitment and to learn quickly of readers' difficulties).

This guarantee covers only advance payment sent in direct response to an advertisement in this magazine (not, for example, payment made in response to catalogues etc, received as a result of answering such advertisements). Classified advertisements are excluded.

NEWS



Microdrives get into first gear

Stephen Adams takes the top off the Sinclair mass storage system

SINCLAIR has at last launched the long-awaited Microdrive. To use it two boxes are required, the Microdrive and the Interface 1. As a bonus the Interface 1 also contains a full handshake-driven RS232 port and a network interface which will allow up to 64 Spectrums to talk to each other.

The Microdrive is the same shape and size as the original Microdrive in advertisements except that the cartridge is now about half the size. The drive is more like a very high-speed tape recorder than a disc drive, as only serial and not random access is allowed.

There is only one moving part in the drive, the motor. It pulls round the tape in the cartridge via a pinch-wheel system at the side of cartridge. The tape has two tracks, each picked up by the stereo tape head fixed inside the box. There is no adjustment needed on the tape head to align it, as two spring clips on either side of the head bring the front of the tape cartridge to the correct position across the head. The cartridge is only 1¼ × 1¼ × 3/16in. and consists of a continuous spool of 20ft. long, 23 microns thick video tape, 1.9mm. wide, made by Thorn-EMI. The tape takes 3.5 seconds to travel one revolution and with two tracks recorded on it has an average access time of 3.5 seconds.

The amount of data stored is 85K minimum per cartridge, giving 680K on eight drives. The cartridge has a write-protect tab which can be removed with a small screwdriver and cannot be inserted into the drive the wrong way round. The entire cartridge is in a thick black case to protect it when not in use. That must be removed before inserting it into the drive. The Microdrive has no dust protection, as its slot is open all the time a cartridge is not in use.

The cartridge can be removed at any time from the drive so long as the drive is not running — only one Microdrive can run at a time. That is indicated by a red LED on the front of the drive.

The drives are inter-connected by a 16-way edge connector with a keyway at pin 3 on each side of the drive and an inter-connecting plug between each. Up to eight drives may be connected to one interface. The connection to the interface is done by a flexible cable and two insulation displacement connectors, approximately 6in. The connections between the drives are secured by a plastic plate screwed between the underside of the drives.

The Interface 1 plugs into the back of the Spectrum via the edge connector and rests underneath the Spectrum, raising it to a 20-degree angle. That is the only connection to the Spectrum and the interface draws all its power through it for the drives and itself. The expansion interface is reproduced just below the original interface, which is covered by a hood of black plastic and all the connections are the same.

The connections to the RS232 are via an Atari-type 9-pin socket at the back (and consist of TX data, RX data, DTR (input), CTS (output), +9V and 0V. A 25-way 'D' plug on a cable is available for £14.95 inc VAT. The speed can be set at anything from 19,200 baud downwards and is programmed by poking a 16-bit number into a new system variable.

The network part of the interface allows up to 100K baud of data to pass over an audio cable 2ft. long between Spectrums. That is supplied with the interface. The terminating jack plugs are 3.5mm. — the same as for the cassette leads. The unit was demonstrated by transferring a screenful (6K) in three seconds down the network.

The control is via extended Basic commands included in an 8K ROM in the unit which is bank-switched in the 16K ROM area.

The Interface costs £29.95 when bought with a £49.95 Microdrive and £49.95 on its own. Postage adds another £4.95. The cartridges cost £4.95 each. Only those notified by Sinclair may order Microdrives — a maximum of two per customer — with as many cartridges as you want.

NEWS

Camel range expanding

THE CAMEL range by Cambridge Microelectronics has been expanded by three new ROM-type packs for the ZX-81. ROM-81 allows you to plug-in up to 8K of EPROMs into the two sockets provided — 2716 or 2732 types.

The memory area covered is selected by soldered straps inside the black plastic box.

The Dream 81 has 64K of RAM plus an EPROM socket which will take up to 16K of EPROM — 27128. The EPROM replaces the RAM between 8K and 16K on the memory map and permits the use of the slower 450ns EPROMs.

The 16K of the 27128 is split into two 8K areas switched in and out by a switch inside the pack, which makes it a little awkward as it is under the cover and not removable when using the machine.

The Cramic is a bigger version of the Memic (16K) which resides in parallel with the 16K RAM on a ZX-81. It can be used to capture and retain any 16K program. Used in conjunction with a 16K RAM pack it allows you to restore a program in seconds. The Cramic is bank-switched by a software I/O output instruction to ports 16 to 31.

The control of switching and copying of memory to the Cramic is done by a small machine code program in a REM statement at the beginning of each program and must be typed-in or run in from TAPE before using the pack.

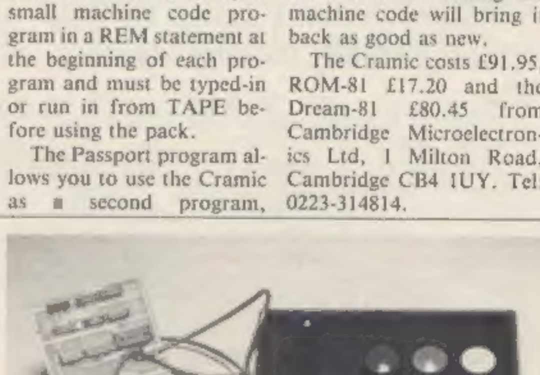
The Passport program allows you to use the Cramic as a second program, switching between that and the original. Spare memory in another location will be required to swap variables between programs.

The Cramic is housed in an 8¾in. × 1in. × 3in. black box which is attached to the ZX-81 by a flexible ribbon cable. An expansion connector is also provided at the back for the RAM pack.

The RAM consists of 6116-type memories backed-up by a lithium battery. The cover has two switches; SEL brings in the Cramic during a program and ON allows the Cramic to be put in parallel with the existing 16K RAM so that any program typed-in may be 'captured' just by de-selecting the Cramic. Unfortunately that crashes the program in RAM but re-inserting the machine code will bring it back as good as new.

The Cramic costs £91.95, ROM-81 £17.20 and the Dream-81 £80.45 from Cambridge Microelectronics Ltd, 1 Milton Road, Cambridge CB4 1UY. Tel: 0223-314814.

Gaming aid



ELECTROTECH has produced a large box containing a programmable joystick and three large push-buttons. In appearance it looks very similar to the large games machine controls. A printed circuit board which contains a 2K RAM plugs into the back of the Spectrum. It is used to store the information on what keys to operate when the joystick switches are operated.

There are eight positions on the joystick as the corners also operate both switches. Whether that will be useful or confusing will depend on the game. All the microswitches are large professional types which should last a long time. That also explains the cost of the joystick, expensive at £43.70 for the standard model.

A tape which accompanying the joystick is a Basic program which allows you to program the joystick and SAVE the results on to tapes as a machine code file. It allows you to re-load the key combinations for a game without the slow process of the Basic program.

The keyboard is not affected and can be used as well as the joystick for entering the score. Seven functions can be programmed — three switches and a four-position joystick. The cost might suit some shops which wear out joysticks very quickly but there seem to be cheaper alternatives for the average user.

Electrotech is at 2 Heath Close, Winston Hill, Luton, Beds. Tel: 0582-429809.

Seeking to convert analogue to digital

I AM a physician interested in computers and owner of a Spectrum. I intend to subscribe to *Sinclair Projects*. In addition, I am interested in literature about building an analogue-to-digital converter and connecting it to the Spectrum, as well as using the machine to control switches.

I know much literature exists for the Vic-20. I would therefore appreciate if you will be able to refer me to the literature about A/D conversion and controlling switches with the Spectrum.

Dr Ron Leor,
Jerusalem, Israel.

● *Issue one of the magazine had two projects, the Latch-Card and Power-Card, which, when used together, allow a ZX-81 or Spectrum to control four relays; a second one could be added, giving control of a total of eight relays.* Sinclair Projects *for June/July, 1983 had an article titled Frequency Gauge which gave some circuits for A/D conversion while the April/May issue contained a full article on a multi-channel A/D and D/A board.*

Graphics

WITH REFERENCE to issue one of *Sinclair Projects*, specifically the Graphics Generator, I wonder if you could supply me the necessary correct circuit diagram and other information as, after two non-working attempts, I would rather the third one works.

Andrew Granger,
Sheffield.

● *Sorry you had difficulty. You could refer to issue two, page 49, which gave the Vero layout, and issue three, page 16, which had some additional notes.*

Making sound

I READ with interest the Spectrum sound generator article in the June/July issue. I wish to build the project for a ZX-81 and would be pleased if you could supply advice for the conversion.

Will the part addresses need to be fully decoded as in the EPROM Programmer for the ZX-81? Can the same parts be used?

Glyn Whatmore,
Ellesmere Port,
Cheshire.

● *If you build as recommended in the June/July issue and incorporate the recommendations from this issue it will work with the ZX-81. You must remember to use the shorter edge connector. As it will be I/O mapped you will need to write a machine code program to control it. More information will be given in future issues.*

Output details

I WAS ABOUT to write suggesting you include an article explaining the various connections on the output connector when I saw the article by Ian Mellor. My delight was short-lived, however, because having ploughed through the first page fairly easily, I was bogged down trying to unravel the explanation offered about how the decoding is performed.

I think Mellor has a good deal of detail logged in his memory which to him makes perfect sense because he can cross-reference every detail within his own memory — but forgets that we humble electronics buffs do not have so much knowledge of the intricacies of computers.

His shorthand way of explaining the decoding left me baffled. He made statements which seemed to refer to nothing he was explaining at the time, expecting the reader to understand what it meant. Starting from the sentence "Each line is scanned eight times", I was lost. He did not explain what a character matrix is, what scanning means, and before all that, why it is necessary to turn off RD and MREQ or what they are, what the lines D0 and D4 are used for, what the refresh register does, what is an interrupt enable?

As you can see I am confused. Perhaps another easier-to-follow article, or should I have done some homework before attempting to follow the article? I am sure many readers will, like me, be attempting to follow the wording with just a basic electronics knowledge and binary theory, and yet wanting to know for what all those connections on the back can be used. Figure 7L does not help much, either. Perhaps his next article will go into more detail.

M O'Toole,
7 Francis Street,
Leeds.

● *Apologies for the difficulty you had with the article. Micro computers, even though they may be small, are really very complex pieces of equipment — even the ZX-81 — and to explain their workings from scratch would fill books. Try the local library for a beginners' book on microprocessors and microcomputers and keep reading* Sinclair Projects. *Most of the articles in* Sinclair Projects *will be at a much simpler level and eventually you will find that they have explained most of the things you did not understand in Mellor's article.*

Video signal

FIRST, congratulations on a superb magazine. Could you tell me whether you can take the video display direct from the Spectrum, via the video channel on the bottom row of the edge-connector? If that is possible, it would save literally thousands of pounds for amateur TV enthusiasts, eliminating the need for a video camera. A monitor — a normal TV set connected to the Spectrum would do.

David Harrison
Burgess Hill,
West Sussex.

● *You misunderstand the video signal. The signal*

LETTERS

from the edge connector pin 15b is that which appears on the TV screen but before it has gone through the modulator. On the issue one machine some links are missing and Y, U, V, and video never reach the edge connector.

Joystick

I READ of the project to construct a Joystick Controller with great interest I would very much like to try this project but I have a ZX-81, not a Spectrum. I am a beginner with computers and their working and therefore would be grateful if you could tell me if this project can be adapted for the ZX-81 and how.

Charles Mann,
Aberdeen.

● *To adapt the Joystick project of issue one to work on the ZX-81 instead of the Spectrum should be easy. The connections to the edge connector will need to be connected to the equivalent pins on the ZX-81 connector — see the edge connector diagram at the back of the magazine. The machine code routine used in the Spectrum version is re-locatable, so we can put it anywhere in RAM and it will still work, and a convenient place on the ZX-81 is in a REM statement in the first line of a Basic program.*

Type-in the program in figure one, run it, and enter the numbers from the DATA statements of lines 40 to 60 of the Spectrum program. Delete all but line one and add the other lines from figure two. GOSUB 9000 will then execute the machine code program to read the joystick positions and return them in variables b and c as for the Spectrum.

To alter the zero count of the X and Y joystick positions instead of changing the numbers in the DATA statements of lines 41 and 43 you will have to POKE the new values into 16516 and 16518.

The sixth, 12th, 21st and 23rd numbers of the machine code are 255, which is the port address of the joystick.

Figure 1.

```
   1 REM   TYPE 43 CHARACTERS
              HERE
  10 LET A1=16514
  20 LET A2=A1+42
  30 FOR A=A1 TO A2
  35 SCROLL
  40 INPUT DATA
  50 POKE A,DATA
  60 PRINT A DATA
  80 NEXT A
```

Figure 2.

```
   1 REM   MACHINE CODE HERE
9000 PAUSE 1
9010 LET A=USR 16514
9020 LET B=INT (A/256)
9030 LET C=A-256*B
9040 RETURN
```

Low-cost solution to the RAM wobble

In the first of a regular spot for tips from readers, we publish this advice from Robert Lorenzo of Dagenham, Essex

AT LAST, a ZX-81 RAM wobble solution. It is the cheapest possible at about 15 pence but for most it will cost nothing. You will need about 6 in. of thin insulated wire.

You must first remove the locating key from the RAM connector shown at B in figure two. Then thread the wire through A shown in figure one. Figure two shows what it should look like so far. Then replace the locating key in the connector.

Then twist the ends of the wire together until you can use pliers but it is not essential — shown in figure three. When it is finished, plug in and you can write a program and can at last breathe at the same time.

The wire does not tighten the connector to the board inside the ZX-81 but instead it takes a firm grip to the case of the computer — so tight that the computer can be held up by RAM alone.

Figure 1.

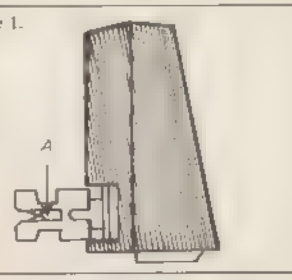


Figure 2.

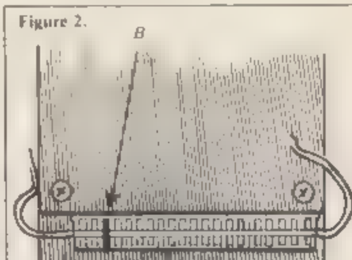
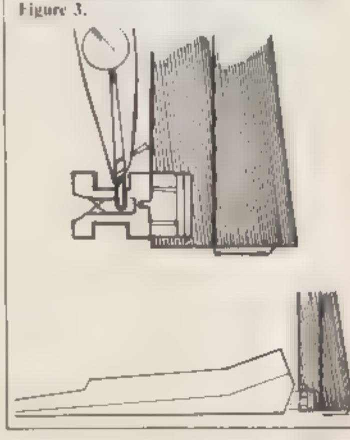


Figure 3.



BATTERY-BACKED RAM

Making sure good routines are stored permanently

Many projects require machine code programs but they have to be entered each time it is set up. Graham Bradley describes a device which allows a copy to be kept.

THE I/O-MAPPED add-ons for the ZX-81 described in this magazine require the use of machine code routines and many will be interested in the speed and efficiency of machine code. With machine code many things can be done which are not possible in Basic.

Machine code can be stored in several places. Above RAMTOP is a good place. It is stored in RAM outside that accessible to Basic. It will not be over-written by a Basic program or erased by NEW. Being independent of the Basic program it is transferable from one program to another. It cannot be saved unless it is transferred to another part of the memory space between 16K and RAMTOP. Its position can be fixed so that non-relocatable code — machine code which includes jumps to specific addresses — can be used.

It can be stored in a REM statement. If a REM statement is the first line of a program it will not be moved about in memory. Basic will not try to execute the REM statement but it will be cleared by NEW and can cause a system hang-up when listed. It is possible to protect the REM line from being listed. The REM line is an integral part of the Basic program and so it can be SAVEd but it is not easy to transfer the machine code from one program to another.

It can be stored in a REM line at the end of a program. It will be EDITable, SAVEable but can be made unLISTable. A suitable place for relocatable machine code.

In the variables area the machine code can be stored as a string array. The position of the array will remain fixed if more than 3+1/4K of RAM is fitted. Use of RUN or CLEAR will erase the machine code. It will be saved with Basic program. To transfer the code between programs it will be necessary to move the array to a position above RAMTOP.

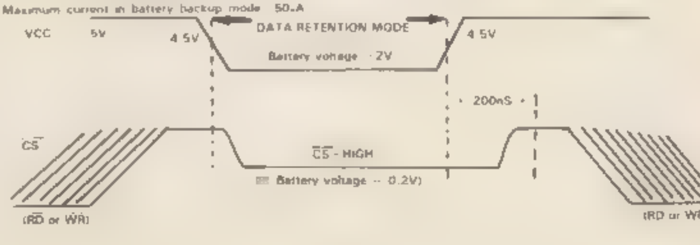
Additional RAM can be decoded to occupy the space between 8K and 16K. That space is transparent to the computer operating system. It is not cleared by a system re-set so that a reset key can be added which is a great help when debugging programs. Being independent of Basic, apart from the initial USER call, it can be transferred from program to program. That area of memory will not be SAVEd and so the machine code will need to be transferred to an array for saving.

The first four techniques can be implemented on the standard ZX-81. The fifth technique requires extra RAM to be added in the 8K to 16K space. Machine code routines often have applications which are useful in many programs. One of the most commonly-used routines is the renumber of Basic lines. Therefore it is useful to have the routines on hand and available to all Basic programs. The battery back-up RAM board acts like an extension to the ROM and the programs in it are always available. It is also useful for transferring data from one program to another.

Its original application was to hold the control programs used when the ZX-81 is dedicated to one task, such as central heating control, weather station or security. The program does not need loading from cassette and the start-up or re-set procedure consists of a single USR call. A later article will describe a small hardware addition which will provide the facility for autostart. With this device a warm re-set will start the program running automatically. Battery back-up provides secure memory which will not lose programs or data if the power fails.

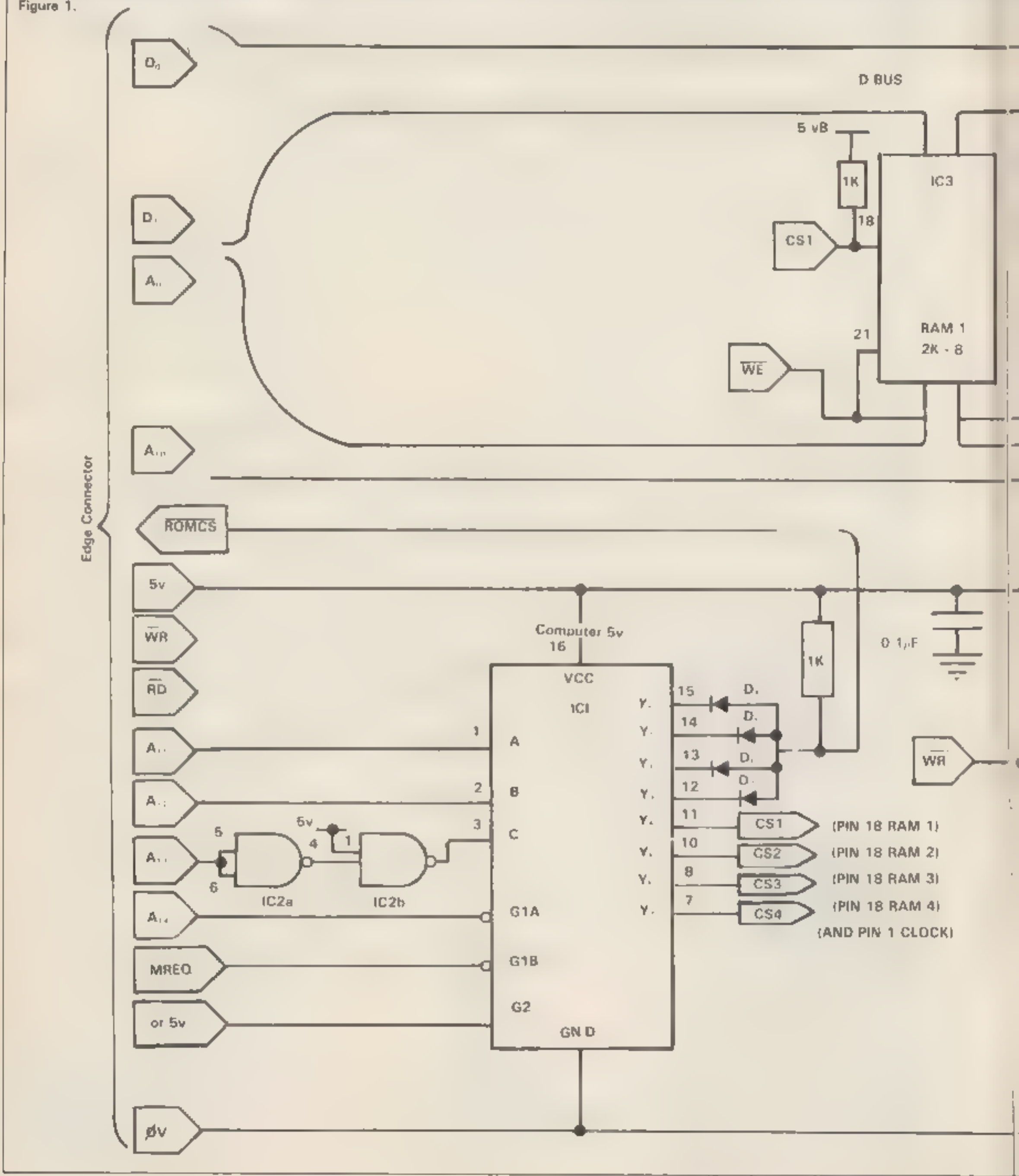
Investigation of figure two the power-down timing diagram for 16K RAM chips, shows that the CS line has to be pulled high to disable the RAM before the RAM Vcc falls below 4.5V. The standby power is supplied from the system through a germanium or Shottky diode. They have a low forward voltage drop of 0.3 to 0.4V. The standby battery is trickle-charged through the 1K8 resistor across D3 which is reverse biased. The electrolytic 63μ capacitor is charged to about 4.7V and is sufficient to maintain the standby power at 4.5V until the system power has

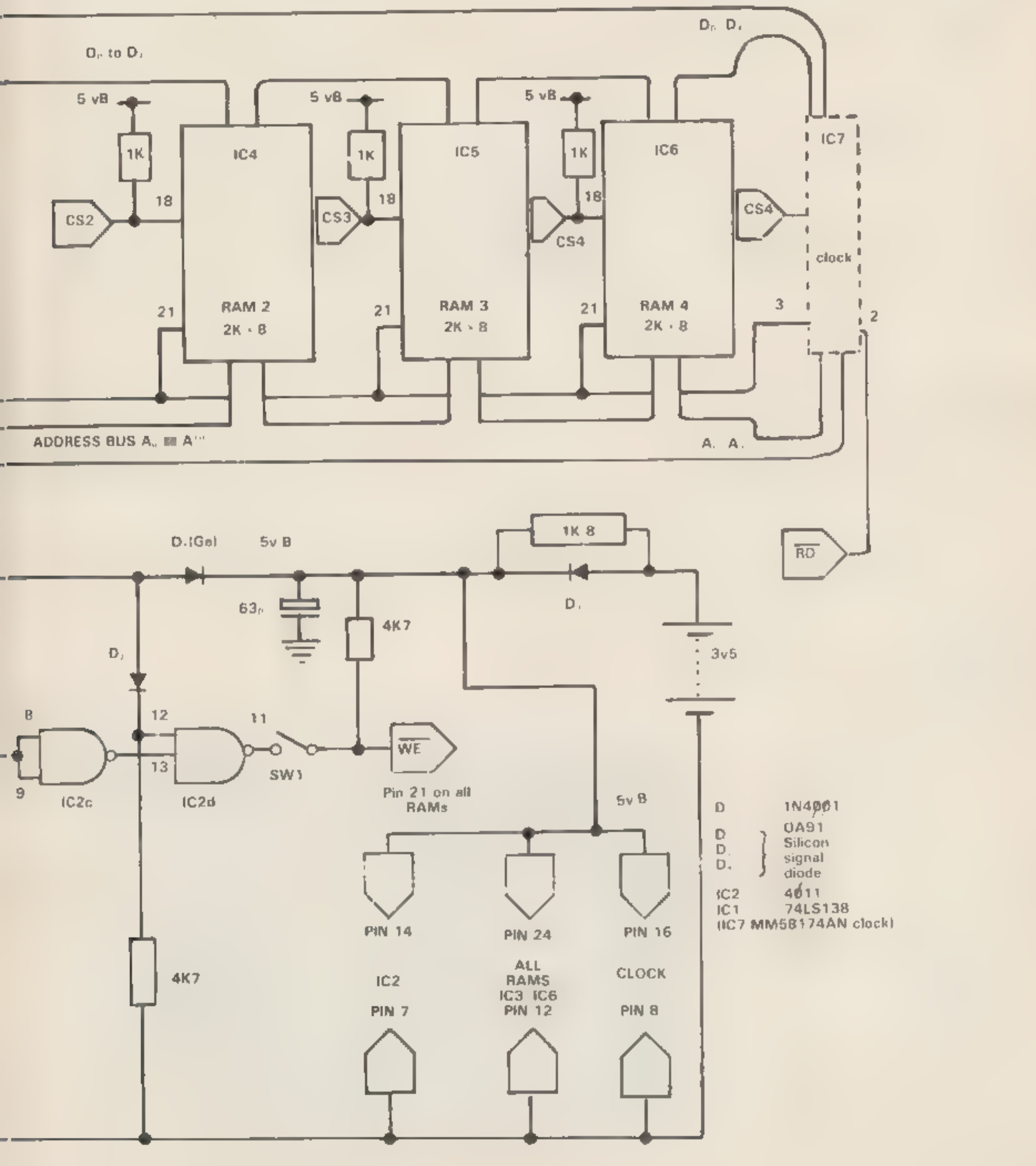
Figure 2: 6116LP data retention character.



BATTERY-BACKED RAM

Figure 1.




D₀ to D₇
D₀ D₇
5 vB
1K
IC4
IC5
IC6
IC7
CS2
CS3
CS4
18
21
RAM 2
2K × 8
RAM 3
2K × 8
RAM 4
2K × 8
clock
3
2
ADDRESS BUS A₀ - A₁₀
A. A.
RD
D.(Ge)
5v B
1K 8
63μ
4K7
D.
3v5
B
12
11
13
9
IC2c
IC2d
SW1
WE
Pin 21 on all RAMs
5v B
D 1N4001
D D D Silicon signal diode OA91
IC2 4011
IC1 74LS138
(IC7 MM58174AN clock)
PIN 14
PIN 24
PIN 16
IC2
PIN 7
ALL RAMS IC3 IC6 PIN 12
CLOCK
PIN 8
4K7

BATTERY-BACKED RAM

Figure 3 CMOS STATIC RAMs

CMOS 16384-BIT STATIC RAMs

- Organized as 2048 × 8
- Address Access Time — 200nS max
- Low Power Dissipation — I^{CC} (Active) = 60mA max

I^{SB} (Standby) = 2mA max

I^{CCDR} (Data Retention) = 10μA max

- Data Retention 2.0V min
- Single +5V DC supply, ±10% tolerance
- Completely static operation, no clocks required
- Equal Access and Cycle Times
- Two Level chip control
- Chip Select
- Output Enable
- Fast $\overline{OE}$ Access Time: 100 ns Max.
- Output timing reference levels: 0.8V-2.2V
- TTL compatible inputs and outputs
- Plug-in compatible with 16K EPROMS
- Industry standard 24-pin DIP package

SPECIAL FEATURES

MB8416 • $\overline{CE}$ (Pin 18) control for simple memory expansion, standby power and data retention

- $\overline{OE}$ (Pin 20) control for fast memory access, output buffer control and elimination of bus contention problems
- $\overline{OE}$ Access Time – 100nS max
- Pin and function compatible with 2716 EPROM
- Pin compatible with HM6116, TC5517, μPD446

MB8417 • $\overline{CE}$ (Pin 18) control for simple memory expansion, standby power and data retention

- $\overline{CS}$ (Pin 20) control for simple memory expansion
- $\overline{CS}$ Access Time — 100nS max
- Pin compatible with TC5516, μPD447

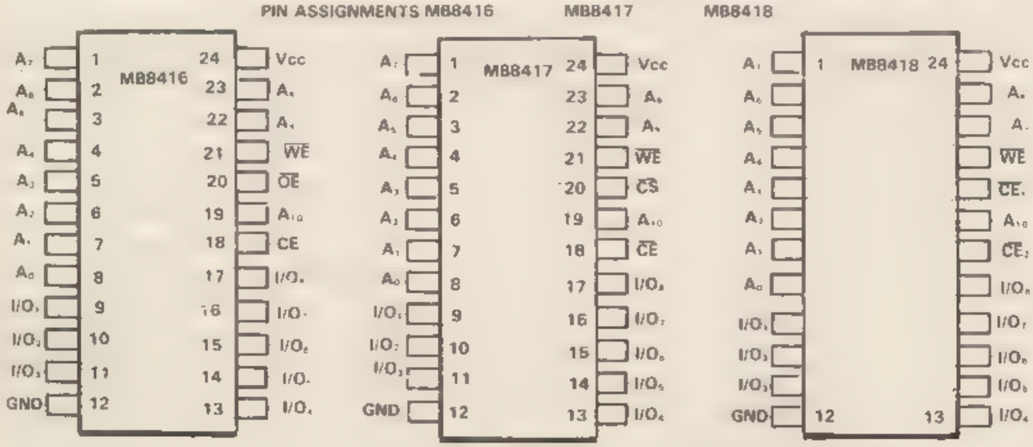
MB8418 • Both CE_2 (Pin 18) and CE_1 (Pin 20) provide power down capability

- $\overline{CE}_2$ and $\overline{CE}_1$ provide simple memory expansion
- $\overline{CE}_2$ and $\overline{CE}_1$ Access Time – 200nS max
- Pin compatible with TC5518

TRUTH TABLE

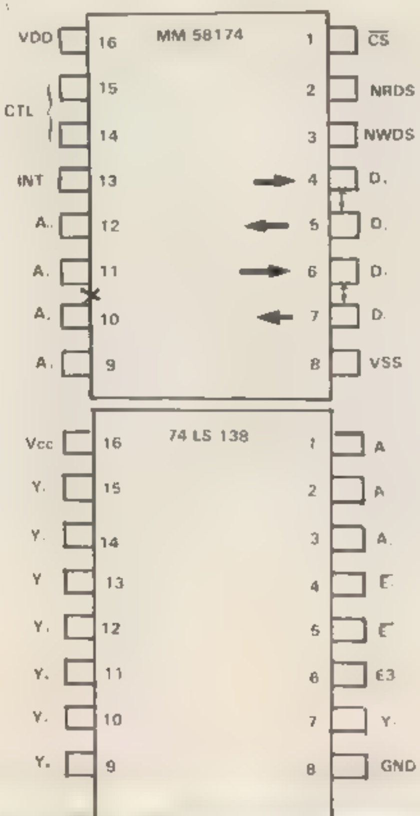
DEVICE NUMBER	MB8416					MB8417					MB8418				
PIN NUMBER	18	20	21	24	9-11 13-17	18	20	21	24	9-11 13-17	18	20	21	24	9-11 13-17
MODE \ PIN NAME	CE	OE	WE	SUPPLY CURRENT	I/O	CE	CS	WE	SUPPLY CURRENT	I/O	CE_2	CE_1	WE	SUPPLY CURRENT	I/O
WRITE	L	X	L	I_{cc}	D_{in}	L	L	L	I_{cc}	D_{in}	L	L	L	I_{cc}	D_{in}
READ	L	L	H	I_{cc}	D_{out}	L	L	H	I_{cc}	D_{out}	L	L	H	I_{cc}	D_{out}
OUTPUT DISABLE	L	H	H	I_{cc}	HIGH Z	–	–	–	–	–	–	–		–	–
CHIP DESELECT	–	–	–	–	–	L	H	X	I_{cc}	HIGH Z	–	–	–	–	–
STANDBY 1	–	–	–	–	–	–	–	–	–	–	X	H	X	I_{sb1}	HIGH Z
STANDBY 2	H	X	X	I_{sb}	HIGH Z	H	X	X	I_{sb}	HIGH Z	H	X	X	I_{sb2}	HIGH Z

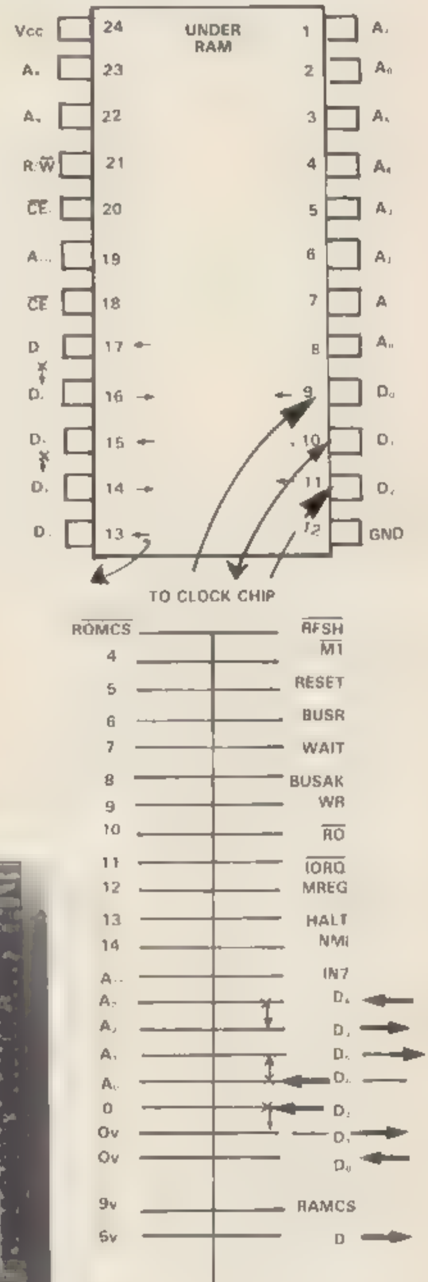
PIN ASSIGNMENTS MB8416 MB8417 MB8418

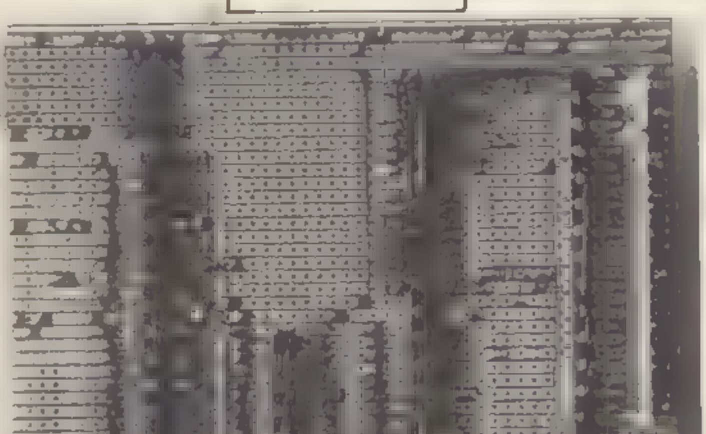


BATTERY-BACKED RAM

Figure 5: Underside view.
Use this as a guide when wiring up







BATTERY-BACKED RAM

fallen to a level where the WE line will be caused to go high.

If you wish to store your programs permanently it is possible to blow an EPROM and put it into the socket in place of the RAM. The 2K EPROMS 2716/2516 are a direct plug-in replacement for the RAM chips used in this project. The 2732 4K × 8 EPROM can be used with a slight modification to the circuits.

The circuit is constructed on the standard size of Veroboard, 36 strips by 50 holes. The address and data lines are taken to at least five sockets, resulting in a great number of interconnections; because of that a wiring pen and the special enamelled wire should be used.

There is room on the board for the addition of the real-time clock described in a previous issue. A real-time clock is very useful in control and monitoring situations. The circuit for the ZX-81 real-time clock will appear in a later issue.

There are four sockets for 2K CMOS RAM chips using either 6116 or 5516 ICs. The fifth socket is for the real-time clock and is decoded to occupy one 2K block of RAM space. A little space and time can be saved by dispensing with either one RAM socket or the clock socket. Two other sockets are provided for the decoding and write enable circuitry.

Insert the sockets into the positions shown and secure them in place by soldering two pins, the power supply pins 7 and 14 or 8 and 16 or 12 and 24. Mount the edge connector so that it is about 6 to 10mm. clear of the board but leave at least 5mm. of the pins on the copper side of the board to allow a ZX-Tongue to be soldered in place to allow further expansion.

Vcc + 5v ± 10%

A_0 - A_{10} — address bus

1/01 - 1/08 — data bus (D0 - D7).

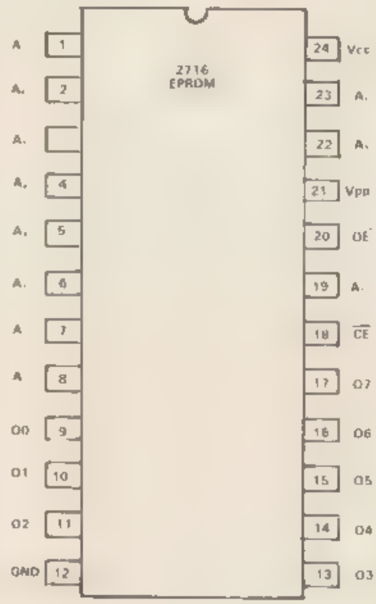
WE — write enable (pin 21) — to WR line

OE — when this pin (pin 20) is high the 6116 data output lock (1/01 - 1/08) is in a high impedance state and the chip cannot be read; connect to low, pin 18 or RD.

CS — this acts like a second chip select or chip erable pin (pin 20 on 5516); connect to low or pin 18.

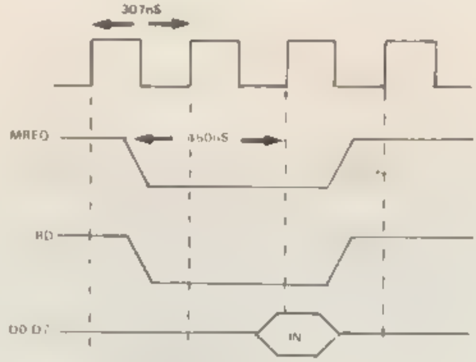
When this high no read or write can take place, RAM is deselected.

Figure 4: 2716 2048 × board [illegible] bits UV erasable and electrically programmable ROM.



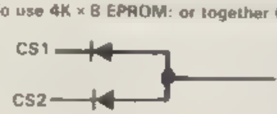
2532	2732
Vcc	Vcc
A_8	A_8
A_9	A_9
Vpp	A_{11}
$\overline{CE}$	$\overline{OE}$/Vpp
A_{10}	A_{10}
A_{11}	$\overline{CE}$
O_7	O_7
O_6	O_6
O_5	O_5
O_4	O_4
O_3	O_3

Vpp use 50mS - 25V pulse when programming otherwise Vpp can be left connected to WE line in battery backup board with no ill effect. The access time with EPROMS [illegible] slower, 450nS rather than 200nS max, but a wait state generator is not necessary



OP CODE FETCH TIMING

To use 4K × 8 EPROM: or together CS1 and CS2



or connect A11 to pin 18 of 2532 or pin 21 of 2732.

BATTERY-BACKED RAM

Secure it in place by soldering the end four pins.

With a track cutter or drill, break all the tracks between the IC pins and between the edge connector terminals. Glue the plastic wiring channel along between the IC pins. Make sure that it is long enough to project beyond the ends of each row of ICs. Insert and solder the wire links.

Place a 4in. high box on the table on which to rest the board so that you are not crouched over while working; that reduces back-ache. Make sure that you have plenty of light and ventilation and try not to inhale the fumes given off by the wire enamel while it is being soldered. The fumes can be dangerous in large quantities.

To melt the special enamel the soldering iron used needs to be hotter than normal or you will not achieve a good connection. Practice on a spare IC socket mounted in a scrap piece of board. Remove the excess solder from a joint after you have made it, then try unwrapping the wire. If it leaves the pin the joint was not hot enough.

Start with the data bus and all the connections to that side of the edge connector. Wrap three or four turns round pin 1, top of the edge connector, take the wire into and along the channel. Take it to pin 17 of the first RAM socket, wrap it twice round the pin then to pin 17 of the second, third and fourth RAM sockets. Without breaking the wire, wrap it around pin 16 (D6) of the fourth RAM socket, then go back to pin 16 of the third, second, first and then pin 7, top of the edge connector.

Again without breaking the connection run to pin 8 top (D5) and wire all the D5 connections (pin 15 on RAM) to the fourth RAM and so on until you have connected all the data bus. Solder all those connections. Break the wires between the pins D1-D2, D5-D6, D3-D4 on the edge connector, and pins D7-D6, D5-D4 on the RAM sockets.

If you wish to add a real-time clock, wire a 16-pin socket as the fifth one in the line but only D3 to D0 and A3 to A0 are used.

RD is connected to pin 2 of the clock chip only, WR is connected to pins 8, 9 of the CMOS NAND gate; MREQ is connected to pin 4 of the 138. At that stage check all the connections for continuity, using a meter. In particular, check all adjacent pins to ensure that there are no short-circuits.

Use the same technique to connect the address bus and finally check that stage of construction by connecting to the ZX-81, switch on, and the cursor should appear. Non-appearance of the cursor will indicate a short circuit due to a link not yet broken, or a solder bridge. I missed breaking the link between A8 and A7 first time.

Operation of the slide switch can cause sufficient movement to crash a program. Once you have decided the configuration of your system, construct a framework to support your add-on boards. Connect the switch by a short length of wire and mount it off the board.

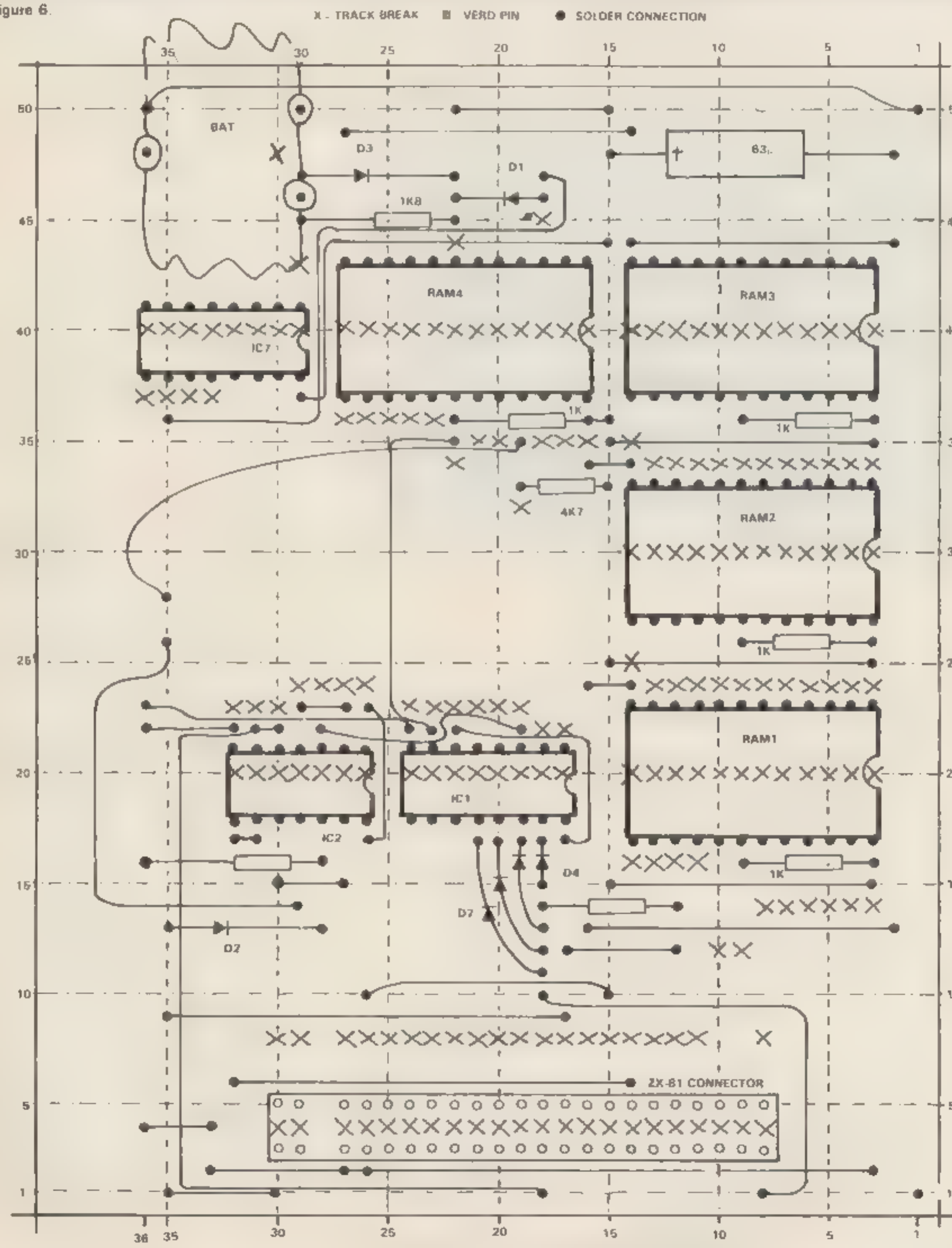
Note that A13 does not go directly to the 74LS138. That is so that it can be driven high by taking the other inputs to the NAND gate high. That feature will be used in a later article describing how you add the autostart feature and the real-time clock.

RD is connected to pin 2 of the clock chip only, WR is connected to pins 8 and 9 of the CMOS NAND gate; MREQ is connected to pin 4 of the '138. At that stage check all the connections for continuity, using a meter. In particular, check all adjacent pins to ensure that there are no short circuits. Use the same technique to connect the address bus and finally check that stage of construction by connecting to the ZX-81, switch on and the cursor should appear.

Non-appearance of the cursor will indicate a short circuit due to a link not yet broken or a solder bridge. I missed breaking the link between A8 and A7 first time round.

Connect the clock chip select pin 1 to RAM 4 chip select. Connect the write enable pins together — pin 21 on RAMs, pin 3 on clock. The chip-select connections are then made to the 74LS138 from each RAM chip. The power and other connections are made using wire links. The slide switch for the write-protect function was glued in place. With the battery and all the ICs the board is heavy and prone to wobble.

Figure 6.


X - TRACK BREAK
VERO PIN
SOLDER CONNECTION
BAT
D3
D1
1K8
RAM4
RAM3
IC7
1K
4K7
RAM2
RAM1
IC2
IC1
D4
D7
D2
ZX-81 CONNECTOR

Getting proper connections makes sure of success

In our previous issue we showed the best methods for organising the layout of components. Peter Grimes now gives advice on how to link the components to ensure a good project.

THERE ARE MANY ways of tackling the assembly of a circuit. Each of us will have a preference for one method. The methods used in the construction of circuits for *Sinclair Projects* are designed so that anyone without the need for special facilities can build a circuit.

Mention is made in this article of some techniques which require special tools.. Those techniques may have certain advantages but it does not exclude the possibility of using Veroboard and wire for circuit construction. The aim is to enable you to build a circuit using your own technique from the circuit diagrams so that it does not matter whether the original circuit was constructed on a PCB or by using wire wrap or any other method.

If a little care is exercised during assembly of electronics projects which are built on Veroboard or printed circuit board, the chances of a correct, reliable, working-first-time project are greatly enhanced.

It was once stated by Sinclair Research that of all the kit ZX-81s returned faulty, about 40 percent were simple faults, such as dry joints which could have been avoided easily with a little care. You cannot send your Veroboard circuit back to Sinclair, so to avoid having a frustrating time searching for dry joints and faulty connections, it is essential to develop a methodical technique.

Figure 1.

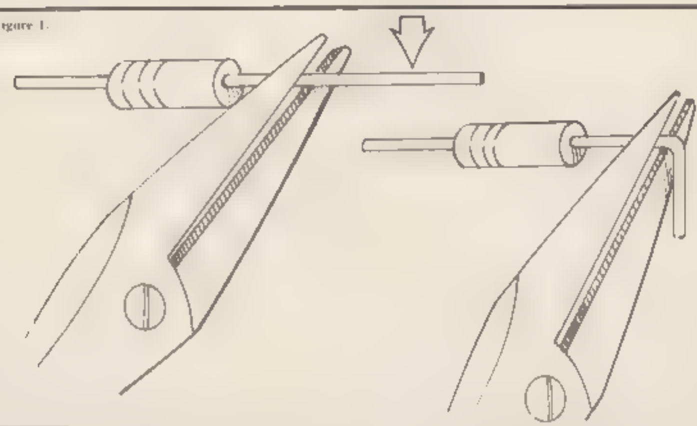
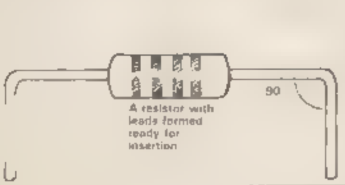


Figure 2.



First, clean the Veroboard; it could be several years old and the copper strips will be oxidised. An ink rubber is ideal. Use something which is not too abrasive—wire-wool is acceptable, even Brillo pads, but wash and dry it well. Glass paper is often too harsh; use a fine grade very sparingly.

Insert IC sockets first and the edge connector. They act as a guide to position the other components. Try to arrange it so that all ICs face the same way; that reduces the chance of putting one in the wrong way round and makes it easier to keep track of pin numbers.

Bend the IC socket pins outwards slightly so that the socket does not fall out when you turn over the board. Preventing the component sliding out can be a problem. Some manufacturers use foam to hold the components in place but I have found that the kind of foam available around the home is not suitable for holding components in place because it melts and invariably somehow finds its way on to the soldering iron.

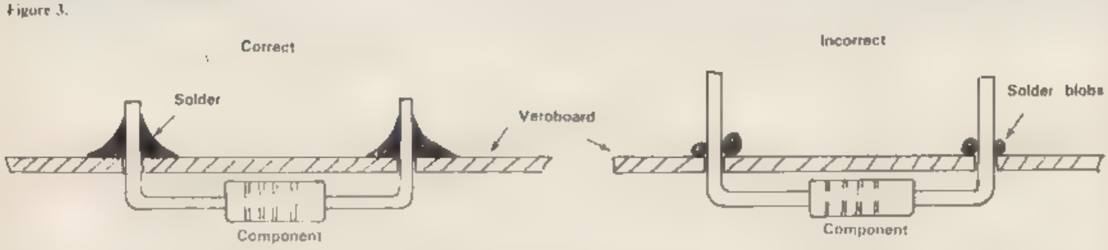
Do not be tempted to bend out the leads too far because that makes it difficult to get a good joint. The usual technique is to insert all passive components, such as resistors, capacitors, IC sockets, first and then solder them all at once. In that way you can check positioning before soldering but if you are careful you can solder as you proceed.

When bending component leads, grip the component side of the lead with a pair of long-nosed pliers—see figure one. A pair of ordinary pliers will do but do not mark or stress the component leads. Then, using your fingers, apply pressure to the lead sticking through the jaws and bend to 90 degrees. The point in doing it that way is so that the body of the component is not stressed at all—hence it is unlikely that cracks will appear in delicate components—and it also makes for a professional-looking job.

Put all resistors with colour codes going the same way. Mount capacitors so that their values are readable. Keep component leads short to avoid short circuits and inductance effects.

WIRING LAYOUT

Figure 3.



Another thing to watch for when inserting resistors into Vero/PC board is to make sure the leads are clean. Often when buying resistors they are in bandoliers which are intended for industrial component-inserting machines and when the paper is removed from the leads a deposit of glue is usually left behind. It is a good idea to remove the glue with a piece of cotton wool soaked in methylated spirit. It is a golden rule when soldering components into place that both surfaces to be soldered should be clean. Cutters can be used to scrape component leads until shiny but do not remove the tinning.

Electronics projects for computers invariably contain integrated circuits. It is advantageous to have every IC in an IC socket, especially CMOS devices, as that facilitates easy removal of the IC, should it be necessary, and avoids damage to CMOS due to static discharge when soldering. One thing to remember when buying Veroboard is that there is more than one type; the type used for most ZX projects has a hole pitch of 0.1in. and will accept dual-in-line sockets.

The size of Veroboard required usually is indicated in the article and we shall be attempting to keep to one standard size of 50 holes by 36 strips. If you are not too sure of the size you need, build the circuit on a large piece of Vero as neatly and compactly as possible and then trim it to size afterwards.

To cut Veroboard score it along a line of holes on the copper side so that you cut the copper strips when trimming across them. Sandwich it between the table and a straight block of wood and break it off. The rough edge can be cleaned with a file. When trimming the other way, score between the copper strips. Determine where the edge connector will be according to the two diagrams and then break all the copper strips which will be between the pins.

We shall be using edge connectors with long wire-wrap-type pins—they are not true wire-wrap pins—so that project boards can be made stackable as shown in figure seven.

Once you have cut the tails, insert the edge connector so that it stands clear of the board by about 6mm. and solder it into place. All other breaks in the track can be made after the components have been soldered into place unless you are using a wiring pen.

When soldering, use a low-wattage iron with a small bit. The correct way to solder a component into place is to tin the bit, i.e., cover the tip of the bit with a thin layer of solder, then heat the component lead and the track on the board at the same time; if that is not done the flux in the solder will flow around the joint, insulating the component lead from the track; that is what is widely-known as a dry joint.

If the joint is hot enough, solder will flow easily around it. A small amount of solder on the iron forms a heat-conducting layer between the iron and the component.

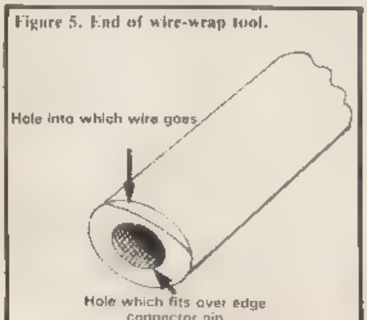
Once the lead and track have been heated, solder is applied sparingly to the joint and when solder flows on to the joint the iron is removed. A good way to remove the iron is along the component lead; that causes a conical shape to be formed, as in figure three, and helps to avoid solder blobs forming around the joint.

It should not be necessary to heat the joint for more than about four seconds at most although beginners may find that difficult. Good soldering is mostly a matter of practice; once learned it is seldom forgotten. One point to bear in mind when soldering semiconductors is that they can be damaged very easily by excessive heat, so it is a good thing if the iron is applied to the joint for as little time as possible.

An ordinary drill bit of almost any size from about ¼in. upwards will suffice for cutting the copper track on

Figure 4. All in one hand wire-wrap tool.

Figure 5. End of wire-wrap tool.



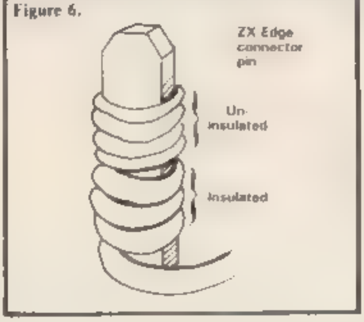
Veroboard. Insert the drill tip into a hole at the joint where the break in the track is needed and then a few turns of the bit will produce a break.

When inserting wire links into circuit boards, choose a multi-core rather than a single-core wire for insulated leads and use tinned copper wire for uninsulated links. That is because multi-core is easier to strip than single-core insulated cable, even with the use of a patent stripping tool. It is all too easy to nick the conductor and when the ends are formed for insertion to the circuit board they usually fall off. So be extremely careful if using single-core insulated wire for links. When using multi-core for uninsulated links, wire strands usually finish all over the place so do not use it for uninsulated links.

When a circuit involves the connection of the data bus to more than one IC it will not be possible to make all the connections with overboard links. First solder all the components and overboard links but do not insert the ICs. Tin the ends of your underboard wire, then tin the track where you wish to attach the wire. If you wish to attach it directly to IC pins, hold the wire hard down against the track near the pin or elsewhere until the solder has hardened. Underboard links do not need to be as short or as close to the board as overboard links. Use sufficient lengths to make it easy work and then press them flat against the board when you have finished.

An easy way of making connection to ZX-type edge connectors is by means of wire wrapping to the edge-connector pins, although some ZX peripherals have edge connectors mounted on small PCB and with no protruding pins. There are various types of wire-wrapping tools on the market. Some of them are very expensive—around £150 for a mains-powered wire-wrapping tool. Just as good a job can be done with cheaper ones which cost about £4. The pins on the better ZX-type edge connectors are a standard 0.84mm. diagonal terminal pin, for which wire-wrapping is made. Some wire-wrapping tools can be used only with certain types of

Figure 6.



wire, e.g., the just-wrap tools available from some of the larger electronic distributors such as Farnell Electronic Components Ltd, Canal Road, Leeds, can be used only with Tefzel insulated wire.

When using the hand wire-wrapping tool, the wire is first stripped by use of the stripper mounted in the handle, as in figure four. The wire is fed through the larger hole in the handle; it then emerges from the smaller hole; if the wire is then pulled out the way it entered it will be stripped for the length which was passed through the larger hole.

The stripped end is then inserted into the hole in the tip—see figure five; typically a length of 25mm. of stripped wire is used. The other hole in the tip is then placed over the pin on the edge connector and the tool rotated about eight times. When the tool is removed from the pin the wire will be wrapped around the pin as shown in figure six. It is important to put about eight or 10 turns on the wrap because three or four turns may not make it a sound joint mechanically and electrically.

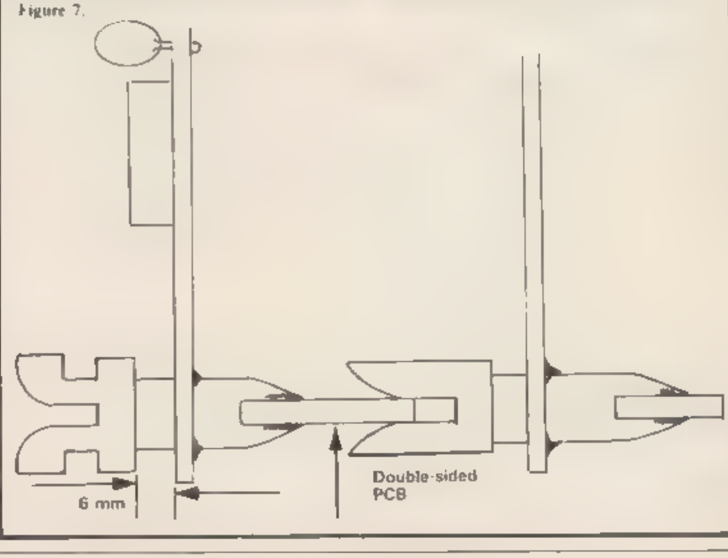
Cable-routeing on circuit boards should be such that all the components are still easily accessible. It is amazing how many beginners route insulated links over ICs and find they have to de-solder a few links to get out the IC.

Push edge-connector pins through Veroboard and solder in place before wrapping. Vero will have to be cut before the connector is inserted because it is almost impossible to do it afterwards.

Wire-wrapping is expensive because special wire-wrap sockets have to be bought which often cost more than the ICs but it has the advantage of high packing density and speed. An alternative method with some of the advantages of wire-wrap is to use a wiring pen.

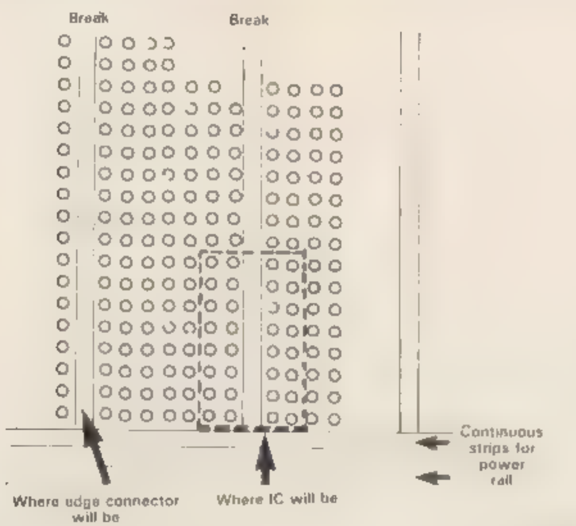
You can use it as a supplement to your usual technique to replace underboard links when a number of data or address lines have to be con-

Figure 7.



WIRING LAYOUT

Figure 8. Preparation of board.



nected. The two most readily-available pens are the Vero-wire pen and the Roadrunner pen.

The special enamelled wire fits in a holder at the top of the pen and passes down through it to emerge from a fine tube at the bottom. The Vero-wire pen has an adjustment to control the tension of the wire. Roadrunner supplies wire with several colours of enamel. You will also need some combs and an iron with a high-temperature bit.

The Roadrunner comb consists of a plastic U channel with dovetail-shaped slots in the sides. That type of comb needs to be glued to the circuit board. The Vero-wire comb consists of a row of cylindrical capstan-shaped pins. That type of comb is held on to the board by short pegs underneath it which fit into the circuit board; specially-prepared stripboards are available for use with wiring pens.

If you are building only a small circuit you can use ordinary Veroboard but it needs preparing first. All breaks in the board will have to be made before you start wiring. Use a straight edge and a hacksaw blade to remove a strip between the pins of each row of ICs and then between the rows of ICs. Keep the IC in line straddling the break. Next insert the IC holders and if they will not stay in place solder on the two power rail pins. The pins should be bent outwards. The combs are then mounted down the centre of each row of IC holders between the IC pins. Evostick clear adhesive works well with the Roadrunner combs. Then experiment until you appreciate the technique.

The edge connector is the most difficult thing to wire. The pen will not fit between the pins and the comb will be no use between them either. Fit the comb on the top side—the data bus side. I bring wires from the address bus through between the pins to the comb.

All the data bus lines will have to be soldered before wiring the address bus, otherwise it will short-circuit against the data bus as the pin heats-up.

Wrap the wire around the pin about three-quarters of the way down and slide it down the rest of the way with the soldering iron. Solder the first connection of a series. Take the wire to the comb and then you have a choice. You can wrap the wire around the dovetail or peg, interweave it, or just run it along the comb.

Too much tension will pull the comb sideways and lose wires; too little will allow the wires to drop out of the comb. Run the wire to the top of the comb where it emerges into fresh air. There is no need for horizontal combs.

Assuming you started from D1 take the wire to D1 on the first IC; give it only two tight turns around the pin. If the turns look like working loose and springing off, anchor the wires around a peg in the comb to keep the tension. Then run to D on the next IC, and so on. That is called daisy-chaining. If the next pin on the last IC is D_1 and D_2 continue to that one, daisy-chaining D_1 and D_2 together; later those lines will be cut.

Work back through all the D_2s to the connector. The next pin on the connector is D_6—there is no need to jump to D_5. Connect all the D_6s. Use the most convenient method for you without losing track of which line you are connecting.

When soldering, slide the wire down the pin with the iron but do not remove the iron at that stage or you may pull off the loop. A hot iron is essential as it has to melt the enamel.

Figure 9.

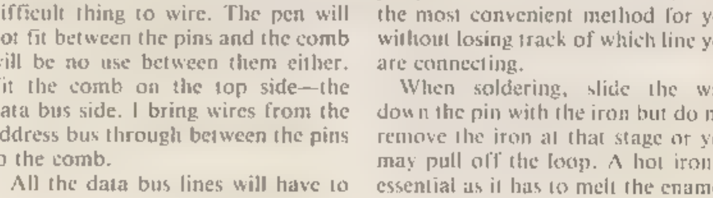
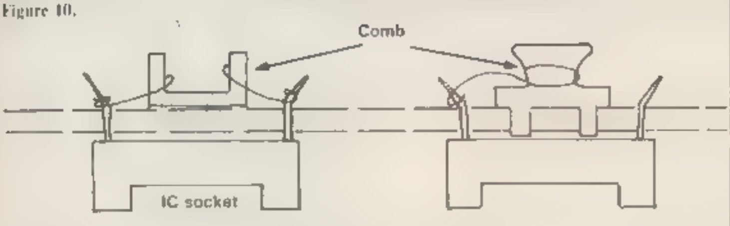


Figure 10.



Use a hot bit on an iron with interchangeable bits, or buy a special iron. Roadrunner sells one for about £6 which has a fine bit.

The enamel gives off a poisonous vapour, so breathe out as you solder and do not bend too close to the work. Do some test joints. Do not put on too much solder and then see if you can unwrap the wire.

Then cut the lines between the adjacent pins. Connect power rails using ordinary wires. Where the wires run between pins, as the address lines from the edge connector do, they can be held in place with Snopake correcting fluid.

The only other common method of wiring we have not dealt with is that of using flexible insulated wire over the top of the board. That is the traditional way of wiring circuits and is often referred to as loom wiring.

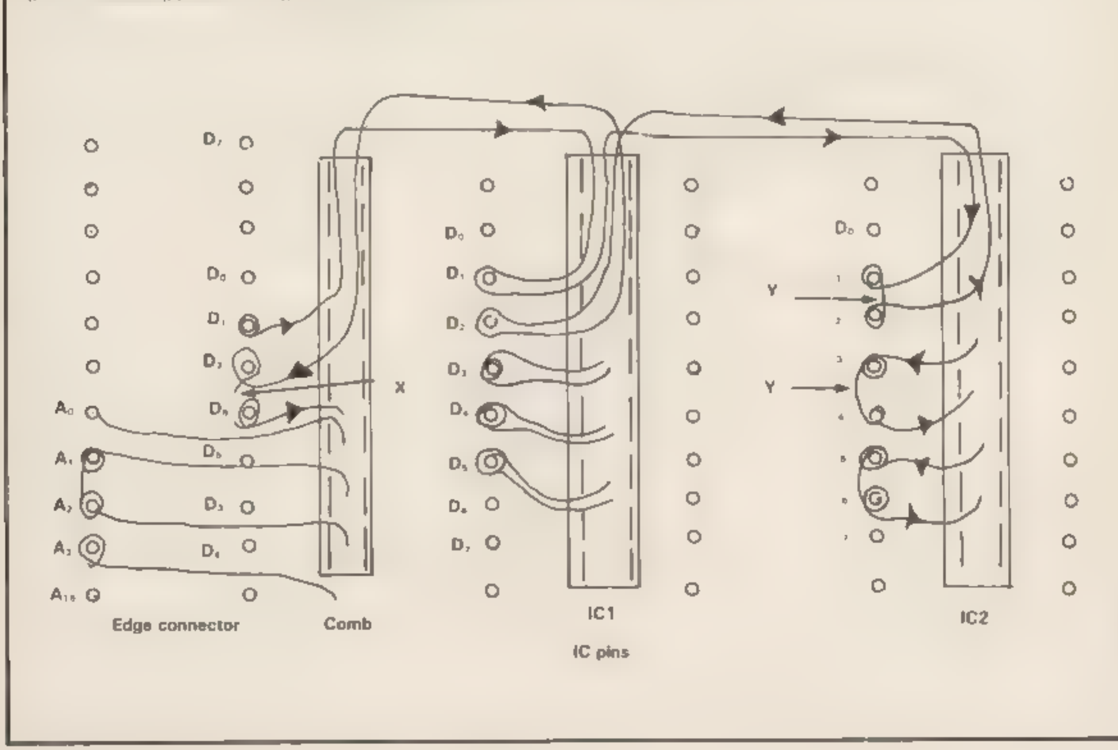
An example of the technique can be found in the Latch Card article in issue one. One end of the wire is soldered into the board at one connection point; the wire is then laid roughly over the board going round ICs, not over the top, and cut to length about 1cm. past the other connection point; that end is stripped and soldered in place.

When all such overboard wires are in place the wires are bunched together with your fingers. For short wiring runs that will be sufficient but with longer runs it is best to tie together the wires and the Veroboard, using cotton thread.

For those who are new to wiring techniques all this may seem a great deal to digest at one attempt, while those who are experts will no doubt decide we have omitted their favourite technique.

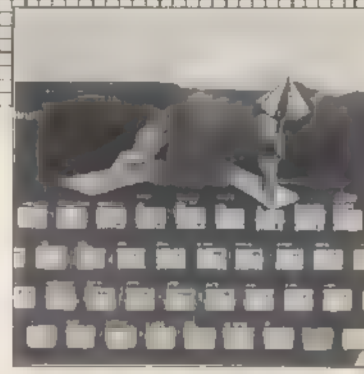
The only way to learn the best wiring technique to use in any particular project is to practise using each one. You do not have to practise on an actual project; you can practise on scraps of Veroboard using IC sockets and a few resistors; that way it does not cost much and you do not have to worry about putting the wires on the correct pins.

Figure 11. Assuming you are wiring, the data bus to two IC's.



WEATHER STATION

Low cost method of keeping record of wind and rain



In the first of a series to build an automatic weather station Graham Bradley describes how to measure wind speed. The complete project will include a number of measurement modules which will work with either the Spectrum or the ZX-81

A DEVICE for measuring wind speed is called an anemometer. The type used most commonly in meteorology consists of three cups pivoted on to a vertical shaft. If they are of reasonable construction the speed of rotation of the shaft will be directly proportional to wind speed. The fact that shaft rotation is directly proportional to wind speed simplifies the software or hardware used in the conversion to wind speed.

There are various electrical techniques for the measurements of rotational speed; a tachogenerator is an AC or DC generator the output of which varies with speed. Commercially-available units have current or voltage outputs which vary linearly with speed. They are expensive. A small model motor or cassette motor can be used; the output will be approximately linear over a limited range. The outputs of tachogenerators and motors will be of the order of one volt at the maximum rotational speeds encountered in anemometers; it will therefore require amplification. At low speeds the output of a small motor, such as a cassette motor, will be irregular and non-linear. That can be alleviated to a certain extent by the use of pulleys and gears to increase the rotational speed of the motor.

The additional friction caused by the tension of the pulley belt requires the use of durable, low-friction bearings. Motors or tachogenerators require careful protection from moisture which can further complicate construction.

The other methods used to measure rotational speed are pulse-counting techniques. The pulses are generated by electromagnetic, capacitive or opto-electronic tranducers or by reed or other switches.

The wind speed is determined by counting the number of pulses occurring in a fixed period. Electromagnetic and capacitive transducers require additional electronics, which may be built into the transducer housing, and are more expensive than opto-electronic or electro-mechanical switch

Figure 1.
Anemometer drives.

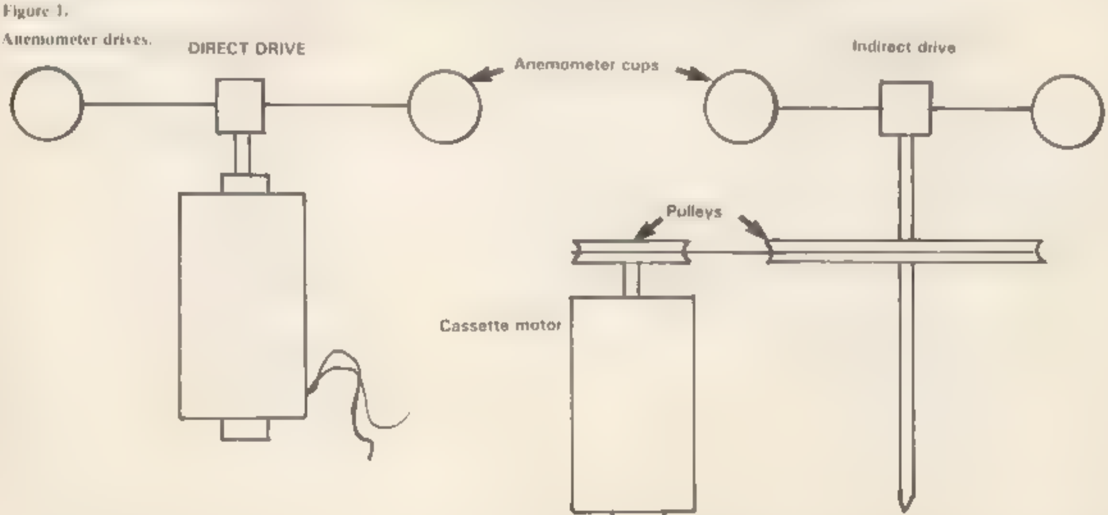
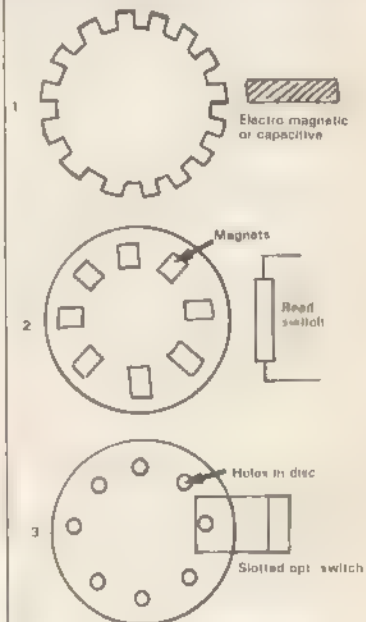


Figure 2. Pulse generation.



methods of pulse generation.

Reed switches are the cheapest and easiest to use. One useful source of suitable reed switches and magnets is certain types of surplus computer keyboards. One reed switch and four magnets should be sufficient. The more magnets you use the more pulses you receive in a given time and the easier it is to measure low wind velocities.

The opto-electronic technique requires a source of light such as a LED and a light detector such as a photo-transistor, photo-diode or photo-resistor. A slotted opto switch simplifies the problem of mounting and aligning the devices. The beam of light will be interrupted by slots or holes cut into a disc.

The main points about the construction of the anemometer are that the spindle should be vertical and the rotating parts well-balanced. The spindle should be able to turn freely to obtain maximum sensitivity and linearity. If a pulse-counting technique is used — there is no sideways force on the spindle — the bearings can be made by drilling holes of clearance size in thin sheet steel. Copper and aluminium are softer and will wear out more quickly. Exposed areas of steel will need to be painted to reduce the rate of corrosion.

The anemometer cups should be free from obstructions which may cause local turbulence and the top bearing should be as near to the top of the spindle as possible.

The anemometer cups can be made from tinplate material soldered into a cone shape or from tennis balls cut in half. A four-cup arrangement is easier to make and balance and uses all four halves of two tennis balls.

If you wish to avoid the difficulty of making and balancing a set of anemometer cups you can use a ready-made plastic anemometer. They have been designed for road safety use but can be modified for measuring wind speed.

The amount of wear on the bottom spindle bearing will depend on the weight of the anemometer cups and attachments. The spindle will require some form of thrust bearing to reduce friction. The simplest method is to file the bottom end of the spindle to a point. An alternative is to find a short tube which is a loose fit on to the spindle and use a single ball bearing as the thrust bearing.

If you wish to avoid the problems of anemometer construction the plastic type is available at a low price — see shopping list.

Production models tend to have a fair degree of friction. The main bearing consists of a single ball bearing in the top of the anemometer central mounting. The ball can easily be lost. Turn the anemometer upside down when separating the two parts. Area D is probably most in need of some sanding with fine emery paper, though areas A, B, and C may also require some attention. The tuned-up anemometer should revolve very freely with the slightest breath of air.

The prototype was made by drilling a hole in the inner part, just above the annular ring D, and a 6.5V, 0.3A bulb was mounted inside the hollow centre. Six holes are drilled in the outer plastic part. The plastic is slightly translucent and some opaque tape can be wrapped round the plastic to increase the light-blocking property of the areas between the holes. Any convenient method of mounting the photo-transistor can be used.

The end of the photo-transistor acts like a lens, so that it is directional. Once aligned it will be unaffected by variations in ambient light level and the output of it varies between less than 0.2V when lit, and greater than 4.5V when unlit, and can thus be fed directly into the input port of a PIO or other device.

The simplest circuit is that of figure six. The signal from the photo-transistor is an analogue one varying linearly with the amount of light falling on it. All such signals should be fed through a Schmitt buffer to remove the uncertainty of voltages between 0.8V and 2.5V. Figure seven shows a Schmitt buffer circuit.

To demonstrate its operation the pulse occurring in a given time can be counted with a Basic routine but Basic is too slow for high wind velocities. A machine code routine can be used to count the cycles (time) between successive pulses.

Figure 4. Spindle bearing.

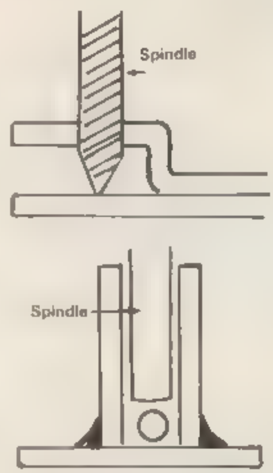
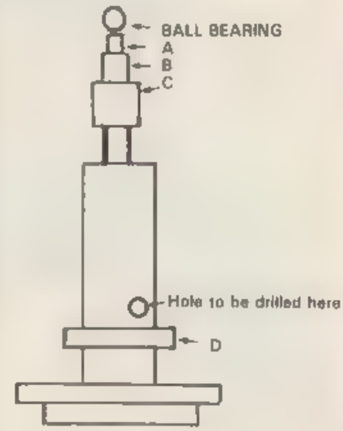


Figure 5. Plastic type anemometer.



The addition of a flip-flop will halve the number of pulses but provides an equal mark/space ratio giving greater reliability to a Basic routine; also the machine code routine could be required to count the time between input level changes.

The in-built mechanical inertia of the anemometer removes the errors usually associated with instantaneous measurements of this type. The time between transitions in the input level will be directly proportional to wind speed.

Figure 3. Anemometer designs.

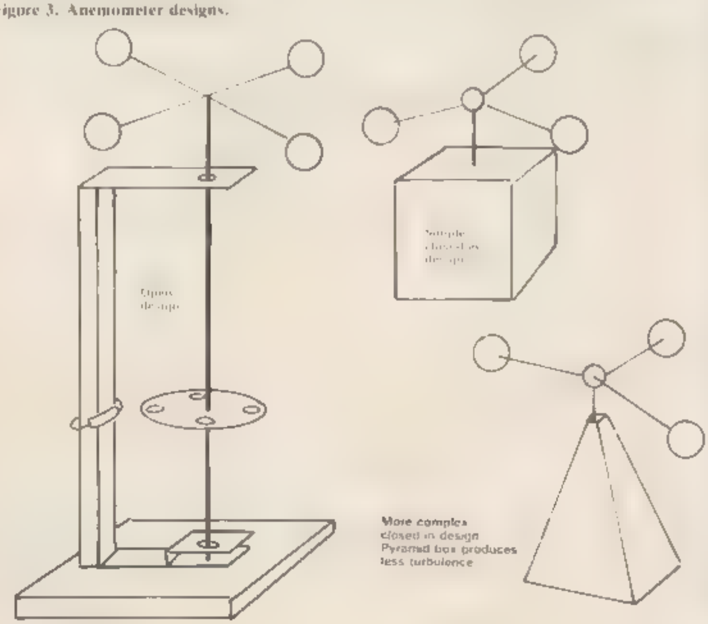


Figure 6.

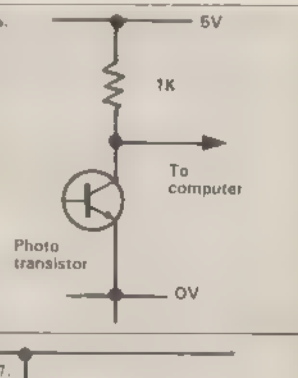
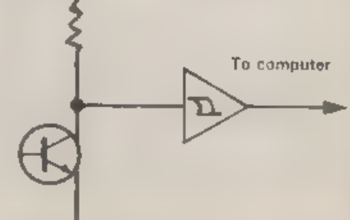


Figure 7.



One method of calibrating the device would be to take the whole set to a weather station,or airport and calibrate it against a professional anemometer.

An alternative method is to build the single chip pulse counter shown in figure eight. Find a long, straight section of private road and when there are no other road users about, and little or no wind, attach the anemometer to the roof rack of the car and travel at different speeds in both directions along the road. The portable pulse counter is calibrated with an oscillator and then a graph plotted to show miles per hour against pulses per second. Reducing the size of C3 to 16nF will increase the range to 0 to 32Hz.

In the next issue we will describe the construction of the wind direction indicator and provide some simple software. If you have suggestions or alternative ideas — either mechanical, electrical or software — please write to the editor.

Figure 9:

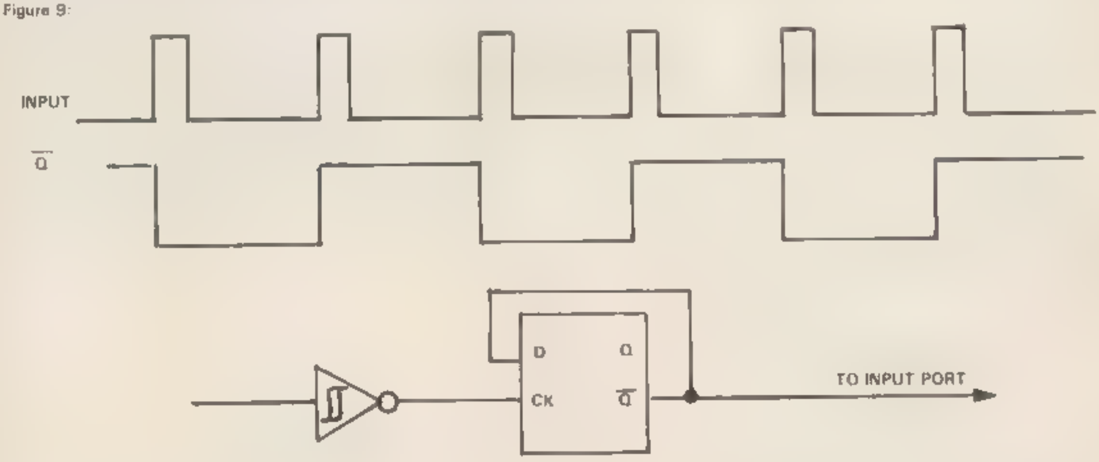
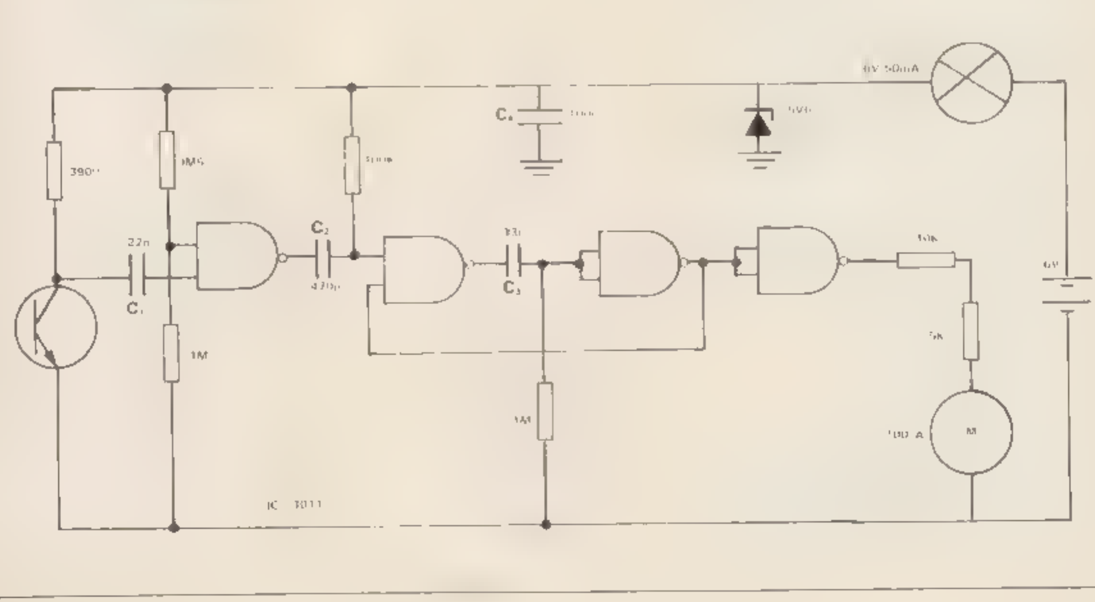
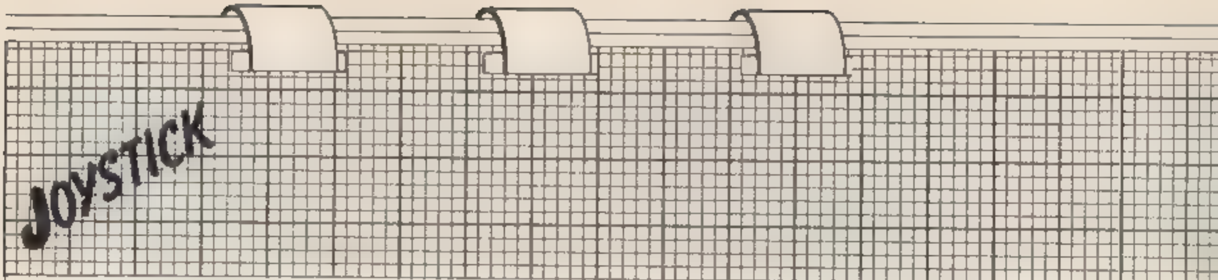


Figure 8: 0 to 16 HZ pulse counter.





Essential peripherals specifies keys to be adapted

Most joysticks are limited in their use by the amount of software specially written to be compatible with them. Corin Howitt shows how a simple project can help to overcome this difficulty

THE MOST ESSENTIAL peripheral to the computer games enthusiast is a joystick, essentially a bank of five or more switches — one for each direction and fire — which usually are configured to mimic specific keys on the computer keyboard. There lies the problem. Which keys should the joystick mimic? The keys which move your laser base left and right in one game may do something different in another, or even nothing at all.

Some manufacturers have attempted to overcome the problem by producing a joystick with a range of compatible software but again you are restricted only to their software, so having a joystick for all your games or other programs could become very costly.

This simple project overcomes that difficulty by allowing you to specify which computer keys are mimiced by the joystick. To understand how the circuit — figure one — works, we need to know a little about how the Spectrum keyboard operates. Figure two shows a simplified version of the Spectrum keyboard circuit.

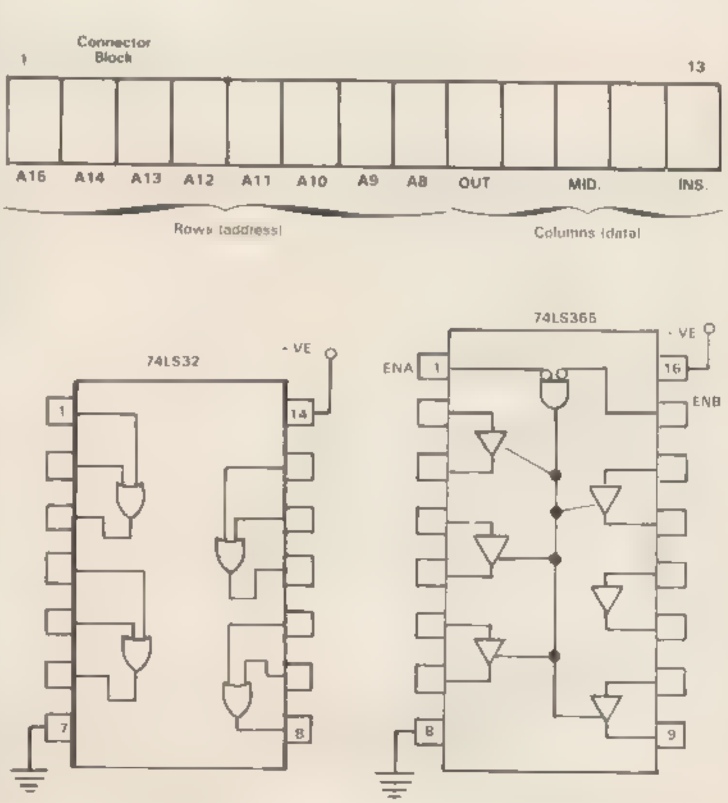
It is a standard keyboard circuit arranged in a 5 × 8 matrix. All the five columns are pulled high by the pull-up resistors R1-5. Each time the monitor program scans the keyboard, about every 20ms, lines A_0 and $\overline{IORQ}$ go low. At the same time, one of the eight address lines A_8-A_{15} goes low to address one of the eight rows.

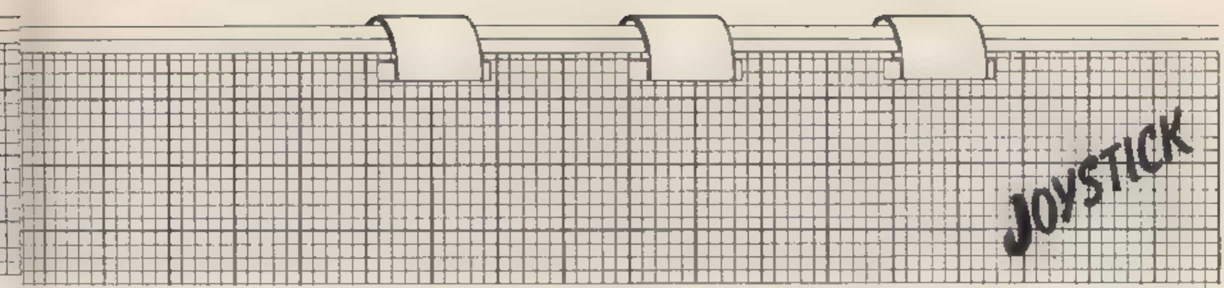
If a key is depressed when its corresponding row address line is low, the row line will sink the column resistor current and the corresponding column line will go low, indicating which key has been pressed. The eight diodes prevent the address lines from being shorted if two keys are pressed simultaneously. To mimic the keyboard all we need to do is to detect when A_0 and $\overline{IORQ}$ go low together and then over-write the keyboard data. R1 and D1 form a discrete component OR gate — the reason for the discrete components is given later. A_0 and $\overline{IORQ}$ are the inputs to that OR gate and the output is fed into the inputs of eight more OR gates, along with the eight row address lines A_8-A_{15}.

Whenever a keyboard scan is made, the output of one of the eight OR gates will go low, indicating which keyboard row is being addressed. A discrete OR gate is used to prevent a third OR gate IC package being used as there are four OR gates in each 74LS32 IC and nine OR gates are required.

The output from the discrete OR gate is also taken to one of the two enable inputs of IC3 a hex three-state

Figure 4.

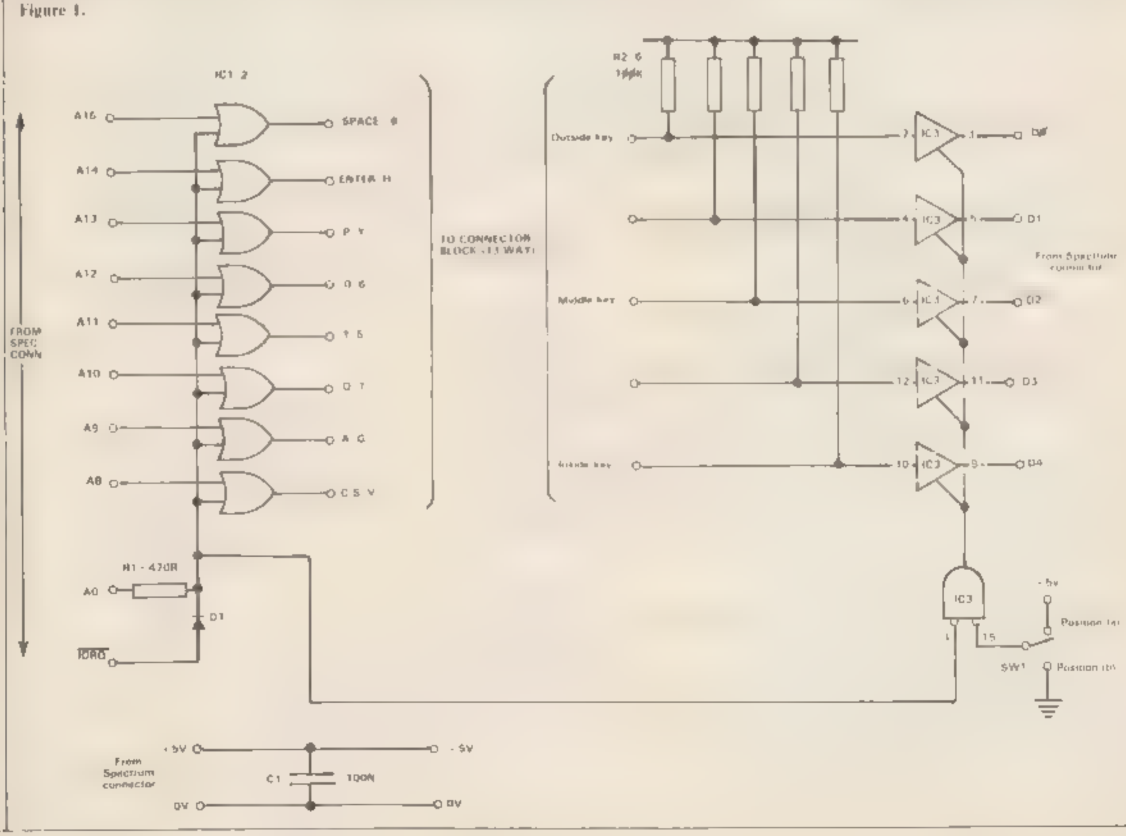




buffer. Five of the six buffers are used and their outputs are connected to data bus lines D_0–D_4. As page 160 of the Spectrum manual indicates, those lines correspond to the keyboard columns. Normally, those outputs will be in a high impedance state but when a keyboard read takes place the state of the buffer inputs will be transferred to the data bus, overwriting any "real" keyboard data which may be present.

Those lines normally will be high due to the pull-up action of R_2–R_6 but if one of the eight OR gate outputs is connected, when it is low, to a buffer input via a joystick switch, the corresponding buffer output will go low, mimicking a depressed keyboard switch. There is a disadvantage. To over-write the keyboard data IC3 has to be a powerful chip, so you should be able to see that if IC3 is over-

Figure 1.



writing the keyboard continuously with highs, the keyboard will not work. That is the function of SW1.

When you need to use the keyboard SW1 should be in position (a). That takes the second enable input to IC3 high, causing all buffer outputs to go into a high impedance state so they cannot affect the keyboard. Placing SW1 in position (b) allows you to use the joystick.

The eight outputs from the OR gates and the five inputs to the buffers are taken to a 13-way connecting or "chocolate" block. To choose which keys the joystick will mimic you must first select the key row — one of the eight OR outputs — and the key position in that row. You then connect the required joystick switch between those two points.

For example, if you wanted the left direction switch on your joystick to correspond to the "Z" key on the Spectrum, you would determine that the "Z" key is in row (a) and in position (b) in that row. That means you would connect the joystick switch between those two terminals on the 13-way connector block.

The project is built on a standard piece of Vero VQ board. Scribe a line using a sharp blade three holes up on one end of the board across the copper tracks. Using a small hammer, knock off that small piece along the edge of a table or bench. That leaves you with two holes from the bottom, allowing you to solder in the edge connector and the wiring.

On the other end of the board, drill two 6BA holes to fix the connecting block. Solder in the edge connector and bolt on the connecting block. Using figure three draw a wiring schedule for the circuit — see issue 1 Latch Card article for an example of a wiring schedule. Solder the IC sockets first, then the discrete components. Next, follow your wiring schedule and wire the circuit. The IC pin-outs are shown — with a suggested pin-out for the connector block as figure four. Then wire in SW1 and insert the ICs. Note that in the prototype a 12-way connector block has been used and hence the inside switch position in each row is omitted.

It is always worth spending time checking your work, especially around the edge connector, as mistakes at that point could prove costly.

Power-down the computer and connect the board, checking to see that SW1 is in position (a). Re-apply power. The copyright message should appear as normal. If it does not, or any other fault appears, disconnect the power immediately and re-check your work.

The first test you should perform is to press the keyboard. It should respond as normal. If it does not, try reversing SW1. If the keyboard still does not work, disconnect the power and re-check your work. Once the keyboard works, reverse SW1. The keyboard should become non-communicative. Any switch-operated joystick such as the Atari or Commodore models will work with this board. The best way to discover the pin-out of a joystick is to test each pin pair methodically with a multimeter switched on its resistance range. As an alternative, some manufacturers produce joystick chassis which can be wired to your liking.

PARTS LIST

- IC1,2 74LS32 quad 2-input or
- IC3 74LS365 hex 3-state buffer
- R1 470R ¼W Carbon (E12)
- R2-6 100K ¼W Carbon (E12)
- C1 100nF Ceramic disc
- SW1 Spot
- 28 + 28-way Spectrum edge connector
- 13-way Connector block, IC sockets
- 2 6BA Nuts and bolts. Vero VQ board
- Connecting wire, solder.

Figure 2. Spectrum Keyboard

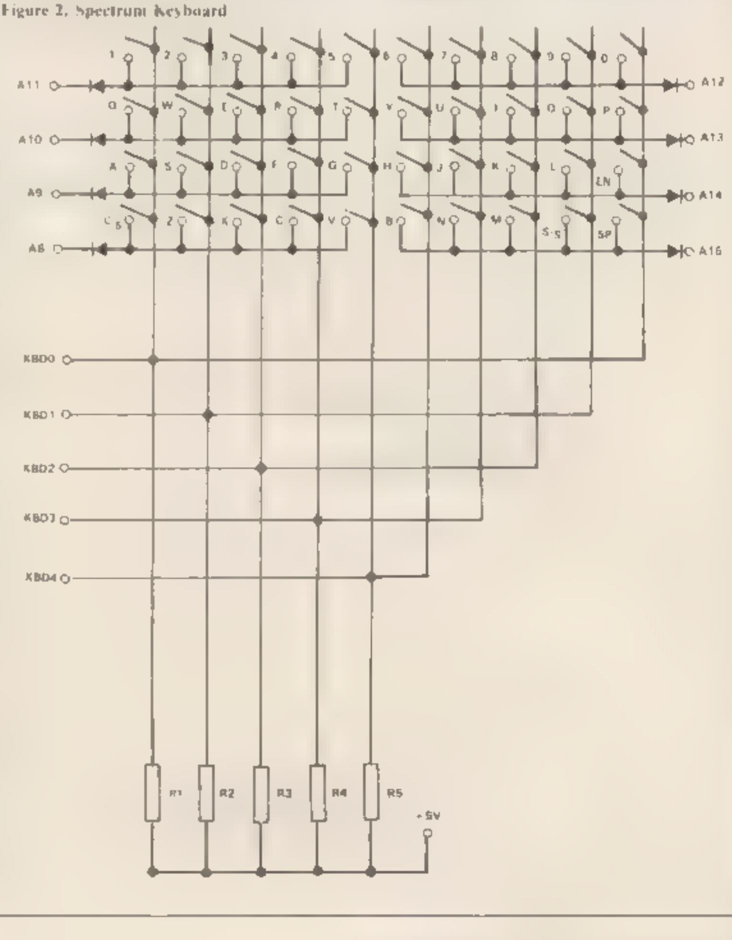
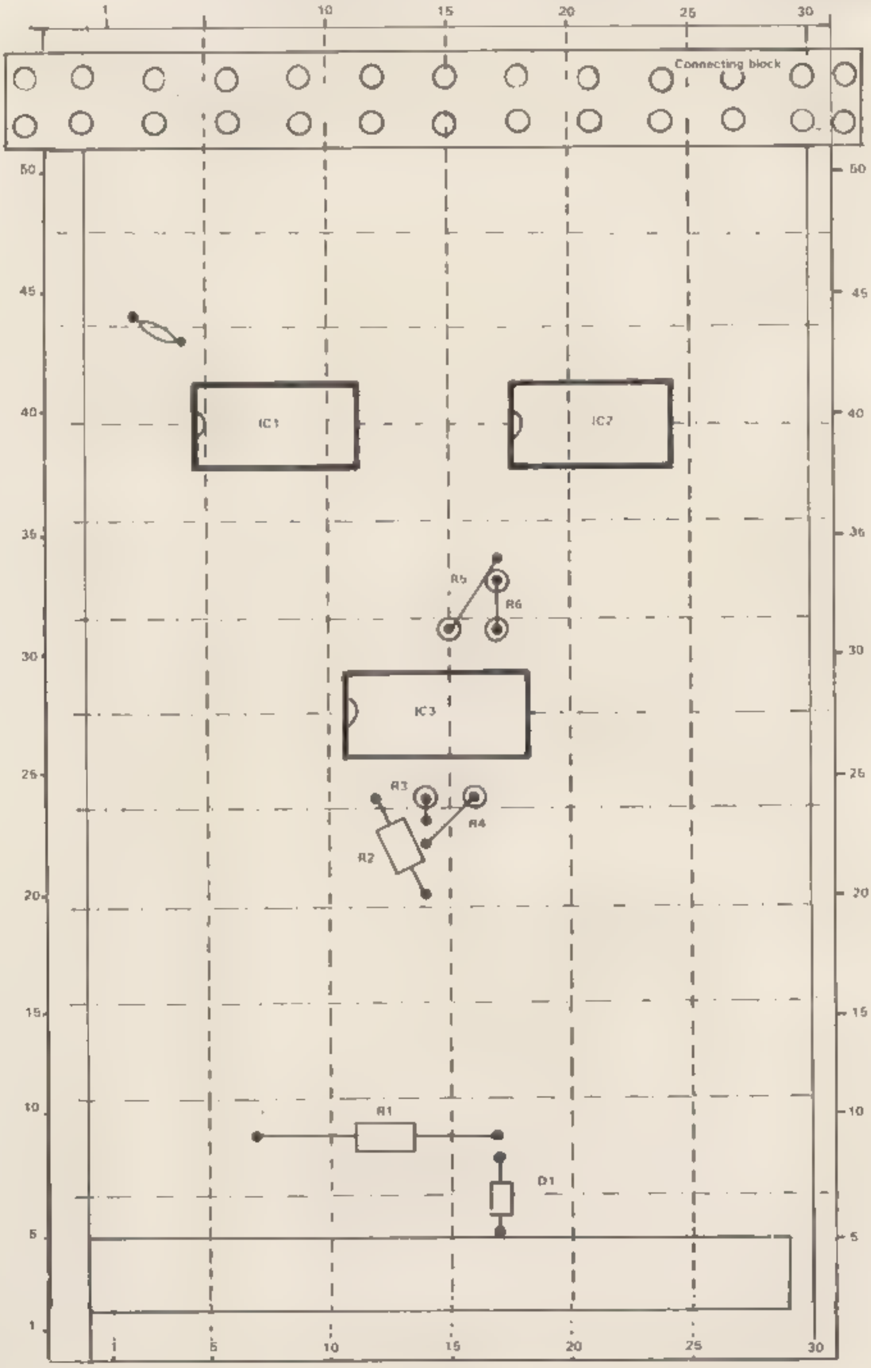


Figure 3. Component orientation. — · — · — Vero VQ track breaks. — — — Direction of tracks.



DESIGN GUIDE

Simulated circuits print-out waveforms

Malcolm Farnsworth shows how a software project can be used to help in the design and analysis of add-ons

A BASIC PROGRAM has been written using a 16K ZX-81 to simulate logic devices and for a given circuit design to print-out a waveform or table of results. It can be used for initial design, analysis or fault-finding and the range of operation is limited only by the memory available.

The information is intended as a guide to permit users to produce their own programs for their particular circuits. Details are given for device simulation and program operation to make it as simple as possible.

Each logic device used must first be translated into Basic statements and figure one shows some common devices with their truth tables and corresponding Basic-derived statements.

The method used is to identify the nodes of a device — input or output — and number them sequentially. Throughout the program, N has been used as the variable for a node. To enable clock waveforms to be produced and also to differentiate between positive- and negative-going clock edges, half-clock pulse periods have been chosen as the minimum time element. The half-clock period variable is denoted by T. An array is chosen DIM(N,T) which will store a time profile of all nodes used in the final circuit.

As an example of a device, look at the NAND gate shown in figure one. The nodes have been identified as a and b for the inputs and c for the output. From the truth table it can be seen that for all input combinations except 11 the output c is 1. Therefore at time T line 1 will cater for all c outputs at 1, while line 2 simulates the conditions when c is 0. That approach has been used for all gates shown and can be extended to multiple input gates as required.

Since flip-flops are clock-dependent they require a little more attention and so let us look at the D type flip-flop. It is characterised by clock input a, data input b, pre-set e, clear f and the truth table is shown. Line 1 tells the output at this time period to stay in the same state as in the previous time period, i.e., no change.

Reference to the truth table indicates that for a positive clock edge the output c will assume the state of the D input and line 2 reflects that statement. Lines 3 and 4 cover the clear and pre-set conditions while line 5 would indicate the invalid state if required. The final line sets d to be the inverse of c.

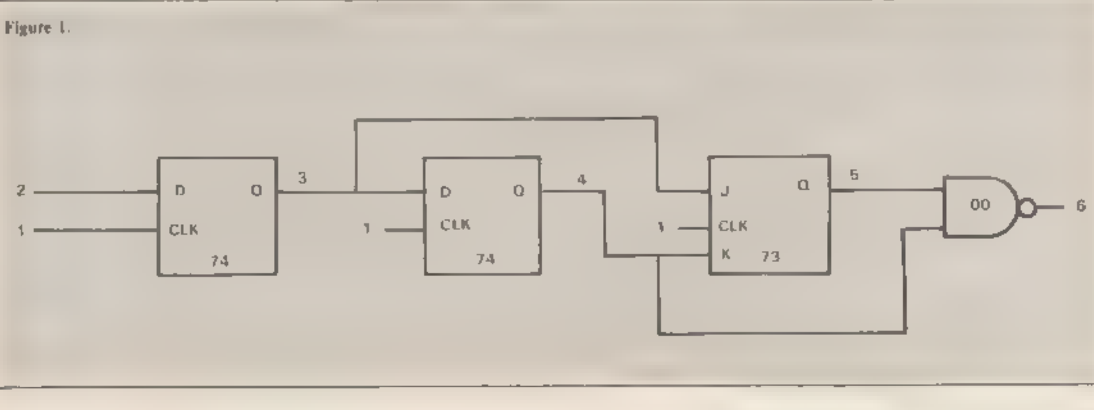
Similar Basic statements can be derived for a J-K flip-flop by reference to its truth table. The methods used for those bi-stables can be extended to shift registers and counters; an example of a binary counter is given. Other logic devices can be simulated by using a combination of the following methods.

Figure one shows a circuit for which a program will be written to indicate a method of use and figure two shows the expected waveforms for a given input.

First all the nodes of the circuit are numbered in logic sequence starting with the clock. Note that all common lines have the same number. Having decided the number of half-clock pulses T required — 12 for this circuit, say — the information and the number of nodes N — six for this circuit — are input at lines 10 to 110 of the program. Line 510 sets up the array used for storing all values of N and T and then sets the array to zero.

It is then necessary to set up and initialise but, with all elements of the

Figure 1.



array at 0, only elements which are at 1 need be identified. At line 800 the clock waveform is set up with all even bits at 1 and line 1000 sets up the input wave form at node 2.

On the first half-clock period — T = 1 — all nodes must be initialised and that will allow the program to run from T = 2 and not invalidate the T − 1 statements used. It has been assumed that all the bi-stables are re-set at time T = 1 and therefore node 6 is the only node at 1 and that is declared in the initialising program at line 1500.

Lines 2000 to 2110 input the node information derived from figure one for each device in turn, which is then looped round for the time period 2 to T. Note that some of the statements derived in figure three can be ignored, depending on the circuit configuration. In that case, since the bi-stables have been re-set initially, no re-set or pre-set information is required and no statement about $\overline{Q}$ is included, since it is not used.

The five statements of figure three for the J-K flip-flop are therefore reduced to three lines — 2060 to 2080.

For the printout an option of waveform or table can be given as at lines 4010 to 4030. To give an idea of what can be achieved, a circuit printout option has also been given. The waveform printout is at line 5000, the table listing at 6000, and the circuit at 7000.

After running the program an option can be re-selected by GOTO 4000. Note that the number of half-clock periods to be selected have been limited to 29 — one screen width — but that can be extended by removing the conditions at lines 70 to 90 and adjusting the print waveform statements. Similarly for the print table, an adjustment of the statements will be required for more than 29 nodes.

Since all the node and time information will have been stored, selected node or time periods can be selected for printout. The printout is done in the SLOW mode but to save time, particularly on bigger circuits, a FAST statement may be needed. Although it may be possible to simplify the listings, to preserve clarity no attempt has been made.

Figure 2.

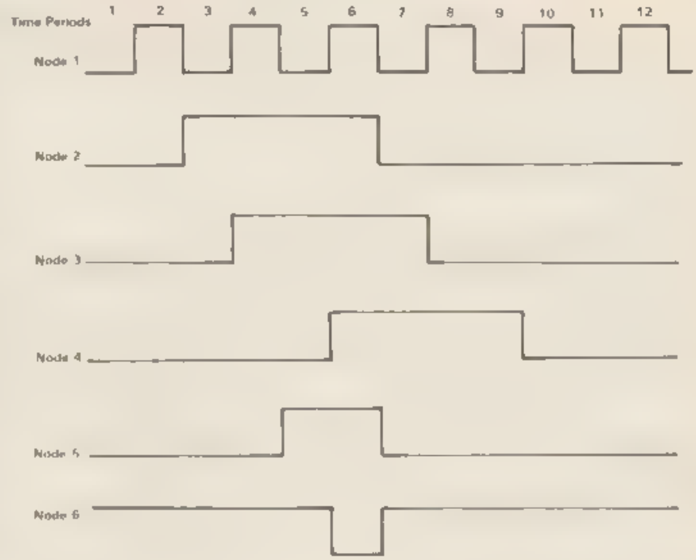


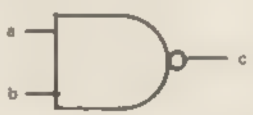
Figure 3. Table of common logic devices

AND (74LS08 type)

a	b	c
0	1	0
1	0	0
0	0	0
1	1	1

1. LET A(c,T) = 0
2. IF A(a,T) = 1 AND A(b,T) = 1 THEN LET A(c,T) = 1

NAND (74LS00 type)



a	b	c
0	1	1
1	0	1
0	0	1
1	1	0

1. LET A(c,T) = 1
2. IF A(a,T) = 1 AND A(b,T) = 1 THEN LET A(c,T) = 0

OR (74LS32 type)

a	b	c
0	1	1
1	0	1
0	0	0
1	1	1

1. LET A(c,T) = 1
2. IF A(a,T) = 0 AND A(b,T) = 0 THEN LET A(c,T) = 0

NOR (74LS02 type)

a	b	c
0	1	0
1	0	0
0	0	1
1	1	0

1. LET A(c,T) = 0
2. IF A(a,T) = 0 AND A(b,T) = 0 THEN LET A(c,T) = 1

DESIGN GUIDE

EXCLUSIVE OR (74LS386 type)

a	b	c
0	1	1
1	0	1
0	0	0
1	1	0

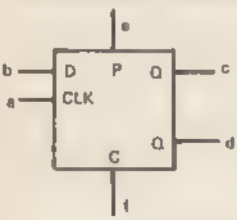
1. LET A(c,T) = 1
2. IF A(a,T) = A(b,T) THEN LET A(c,T) = ■

INVERTER (74LS04 type)

a	b
0	1
1	0

1. LET A(b,T) = 1 − A(a,T)

D TYPE FLIP-FLOP (74LS74 type)

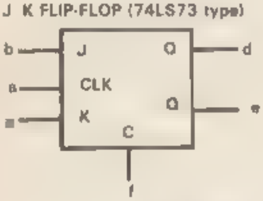


e	f	a	b	c	d
0	1	x	x	1	0
1	0	x	x	0	1
0	0	x	x	1	1 *
1	1	+	1	1	0
1	1	+	0	0	1
1	1	0	x	■	$\bar{Q}$

* unstable state
\+ positive clock edge
x don't care
$Q\bar{Q}$ outputs stay same

1. LET A(c,T) = A(c,T − 1)
2. IF A(a,T − 1) = 0 AND A(a,T) = 1 THEN LET A(c,T) = A(b,T − 1)
3. IF A(f,T) = 0 THEN LET A(c,T) = 0
4. IF A(e,T) = 0 THEN LET A(c,T) = 1
5. IF A(e,T) = 0 AND A(f,T) = 0 THEN PRINT "ERROR"
6. LET A(d,T) = 1 − A(c,T)

J K FLIP-FLOP (74LS73 type)

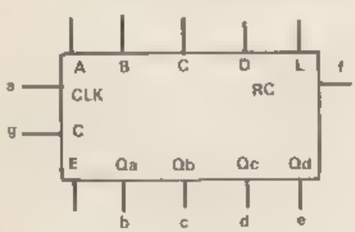


f	a	b	c	d	e
0	x	x	x	■	$\bar{1}$
1	1	x	x	Q	$\bar{Q}$
1	−	0	0	Q	$\bar{Q}$
1	−	1	0	1	0
1	−	0	1	0	1
1	−	1	1	C	$\bar{C}$

x don't care
− negative clock edge
$Q\bar{Q}$ outputs stay same
$C\bar{C}$ outputs both toggle

1. LET A(d,T) = A(d,T − 1)
2. IF A(a,T − 1) = 1 AND A(a,T) = 0 AND A(b,T − 1) = 1 AND A(c,T 1) = 1 THEN LET A(d,T) = 1 − A(d,T − 1)
3. IF A(a,T − 1) = 1 AND A(a,T) = 0 AND A(b,T − 1) = 1 − A(c,T − 1) THEN LET A(d,T) = A(b,T − 1)
4. IF A(f,T) = 0 THEN LET A(d,T) = 0
5. LET A(e,T) = 1 − A(d,T)

Figure 3. Synchronous four-bit binary counter — 74LS161-type positive edge-triggered.



count	b	c	d	e	f
0	0	0	0	0	0
1	1	0	0	0	0
2	0	1	0	■	0
3	1	1	■	0	0
4	0	0	1	0	0
5	1	0	1	0	0
6	0	1	1	0	0
7	1	1	1	0	■
8	0	■	■	1	0
9	1	0	0	1	0
10	0	1	0	1	0
11	1	1	0	1	0
12	0	0	1	1	0
13	1	0	1	1	■
14	0	1	1	1	0
15	1	1	1	1	1

L − load E − enable
C − clear RC − ripple carry
For normal counting Enable is at 1

For initialising Load is 0 and first positive clock sets b to e to values given by A to D.

1. LET A(b,T) = A(b,T − 1)
2. IF A(a,T − 1) = 0 AND A(a,T) = 1 THEN LET A(b,T) = 1 − A(b,T − 1)
3. LET A(c,T) = A(c,T − 1)
4. IF A(b,T − 1) = 1 AND A(b,T) = 0 THEN LET A(c,T) = 1 − A(c,T − 1)
5. LET A(d,T) = A(d,T − 1)
6. IF A(c,T − 1) = 1 AND A(c,T) = ■ THEN LET A(d,T) = 1 − A(d,T − 1)
7. LET A(e,T) = A(e,T − 1)
8. IF A(d,T − 1) = 1 AND A(d,T) = 0 THEN LET A(e,T) = 1 − A(e,T − 1)
9. LET A(f,T) = 0
10. IF A(b,T) = 1 AND A(c,T) = 1 AND A(d,T) = 1 AND A (e,T) = 1 THEN LET A(f,T) = 1

To introduce a reader's circuit it will be necessary to include relevant statements in each section — lines 800 to 2110 — in a similar manner to the example. After establishing the basic circuit, different waveforms or extra circuits can easily be added to establish the circuit response.

```
   1 REM WAVEFORMS      MAY 1983
   2 REM BY M.FARNSWORTH
  10 PRINT "INPUT NUMBER OF CIRC
UIT NODES   (2 OR MORE THEN NEWL
INE)"
  20 INPUT N
  30 IF N>=2 THEN GOTO 50
  40 GOTO 20
  50 PRINT N
  60 PRINT
  70 PRINT "INPUT NUMBER OF HALF
 CLOCK        CYCLES (2 TO 29 THEN
 NEWLINE)"
  80 INPUT T
  90 IF T>=2 AND T<=29 THEN GOTO
 110
 100 GOTO 80
 110 PRINT T
 120 FOR X=1 TO 70
 130 NEXT X
 140 FAST
 150 CLS
 500 REM SET ARRAY TO 0
 510 DIM A(N,T)
 520 FOR X=1 TO N
 530 FOR Y=1 TO T
 540 LET A(X,Y)=0
 550 NEXT Y
 560 NEXT X
 800 REM CLOCK WAVEFORM
 810 FOR Y=2 TO T STEP 2
 820 LET A(1,Y)=1
 830 NEXT Y
1000 REM DATA WAVEFORM
1010 FOR Y=3 TO 6
1020 LET A(2,Y)=1
1030 NEXT Y
1500 REM INITIALISE
1510 LET A(6,1)=1
2000 REM I/P NODES
2010 FOR Y=2 TO T
2020 LET A(3,Y)=A(3,Y-1)
2030 IF A(1,Y-1)=0 AND A(1,Y)=1
THEN LET A(3,Y)=A(2,Y-1)
2040 LET A(4,Y)=A(4,Y-1)
2050 IF A(1,Y-1)=0 AND A(1,Y)=1
THEN LET A(4,Y)=A(3,Y-1)
2060 LET A(5,Y)=A(5,Y-1)
2070 IF A(1,Y-1)=1 AND A(1,Y)=0
AND A(3,Y-1)=1 AND A(4,Y-1)=1 TH
EN LET A(5,Y)=1-A(5,Y-1)
2080 IF A(1,Y-1)=1 AND A(1,Y)=0
AND A(3,Y-1)=1-A(4,Y-1) THEN LET
 A(5,Y)=A(2,Y-1)
2090 LET A(6,Y)=1
2100 IF A(5,Y)=1 AND A(4,Y)=1 TH
EN LET A(6,Y)=0
2110 NEXT Y
4000 SLOW
4010 PRINT "1 PRINT WAVEFORM"
4020 PRINT "2 PRINT TABLE"
4030 PRINT "3 PRINT CIRCUIT"
4040 PRINT
4050 PRINT "INPUT 1,2,OR 3 THEN
NEWLINE"
4060 INPUT B
4065 CLS
4070 IF B=1 THEN GOTO 5000
4080 IF B=2 THEN GOTO 6000
4090 IF B=3 THEN GOTO 7000
4100 GOTO 4010
5000 REM PRINT WAVEFORM
5010 LET A$="N T12345678901234S6
7890123456789"
5020 PRINT A$(1 TO T+3)
5030 FOR X=1 TO N
5040 PRINT X;TAB 3;
5050 FOR Y=1 TO T
5060 LET P=A(X,Y)
5070 IF P=0 THEN PRINT "(g6)";
5080 IF P=1 THEN PRINT "(g7)";
5090 NEXT Y
5100 PRINT
5110 PRINT
5120 NEXT X
5130 STOP
6000 REM PRINT TABLE
6010 LET A$="T N12345678901234S6
7890123456789"
6020 PRINT A$(1 TO N+3)
6030 FOR Y=1 TO T
6040 PRINT Y;TAB 3;
6050 FOR X=1 TO N
6060 PRINT A(X,Y);
6070 NEXT X
6080 PRINT
6090 NEXT Y
6100 STOP
7000 REM PRINT CIRCUIT
7010 LET Y=15
7020 LET X=2
7030 GOSUB 7500
7040 LET X=9
7050 GOSUB 7500
7060 LET X=16
7070 LET Y=13
7080 GOSUB 7600
7090 LET X=23
7100 GOSUB 7700
7110 PRINT AT 16,2;"2"
7120 PRINT AT 16,9;"3"
7130 STOP
7500 REM SYMBOL ROUTINE
7510 PRINT AT Y,X;"(2*sP isP iP·
isP)"
7520 PRINT AT Y+1,X;"--( id·isP
i4 )--"
7530 PRINT AT Y+2,X;"1-(·i<·2*is
P)"
7540 PRINT AT Y+3,X;"(2*SP 3*isP
)"
7550 PRINT AT Y+4,X;"(2*sP isP i
c·isP)"
7560 RETURN
7600 PRINT AT Y,X;"(2*sP·3*isP)"
7610 PRINT AT Y+1,X;"3-(iJ isP i
q·)--"
7620 PRINT AT Y+2,X;"1-(·i< 2*is
7630 PRINT AT Y+3,X;"4-(·ik isP)
"
7640 PRINT AT Y+4,X;"(2*sP isP i
c isP)"
7650 RETURN
7700 PRINT AT Y,X;"(2*sP 3*isP)"
7710 PRINT AT Y+1,X;"5-(·3*isP)"
7720 PRINT AT Y+2,X;"(2*sP isP·i
&·9c·)-6"
7730 PRINT AT Y+3,X;"4-(·3*isP)"
7740 PRINT AT Y+4,X;"(3*sP·3*isP
)"
7750 RETURN
```

All graphic instructions are put into brackets within quotation marks to separate them from ordinary print. The letter 'i' means inverse and the letter 'g' means graphic. The 'sp' marker means space. For instance 6*g4 would mean that six of the graphic character 3 on the '4' key should be entered. The marker 'isp' means inverse space, which is a black square.

Expanding the alarm system to suit your requirements

Corin Howitt describes two projects, both hardware and software, which can be added to the major article in the last issue

HERE ARE TWO add-ons to the low-cost computer-controlled burglar alarm to protect your home described in the last issue. The first is a program illustrating how to develop software for the alarm system and the various functions which can be realised. The second is a hardware design to improve the back-up alarm monitoring for when the alarm is off or otherwise occupied.

The program, shown as listing one, will monitor the input lines of the alarm for either a 1 or 0 as defined by you and will produce an alarm if an error is detected. You can also instruct the program to ignore one or more input lines so that you can switch each line in or out of the system. Further, the program will generate a 30-second delay on exit and re-entry — both delays are adjustable independently — and will ignore an alarm being generated on any lines within the entry/exit route.

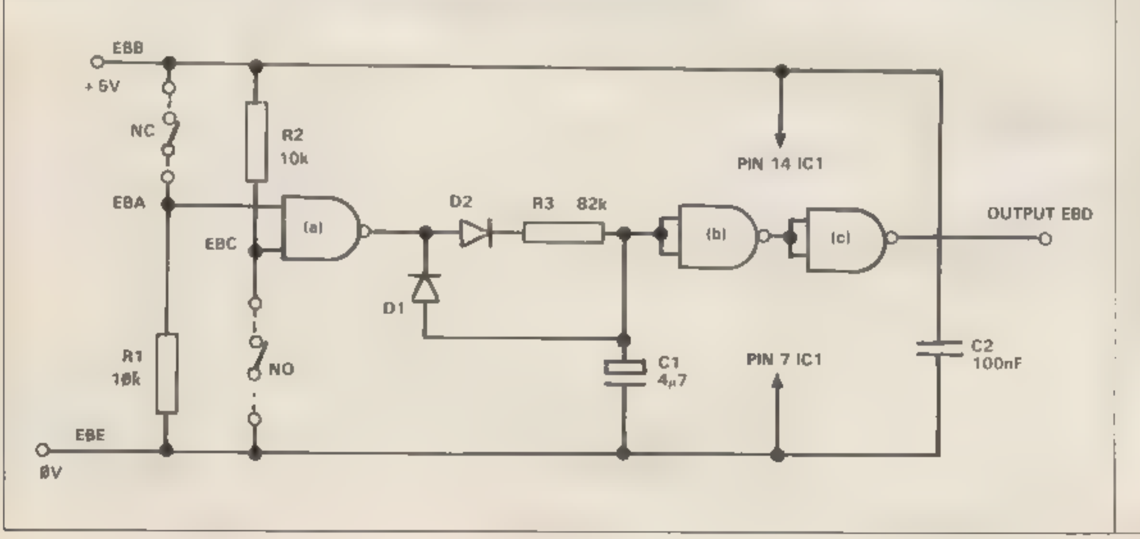
The program works by inputting information from the user and storing the information in eight element arrays, corresponding to the eight data lines. The first four lines initialise those arrays, make the keys bleep, and set the screen colours. The latter two can, of course, be changed.

Next follows a series of FOR-NEXT loops. The first, starting at line 50, requests whether a line should be NO or NC. If the line is to be NC, the value on the line should be a 1 and a 1 is stored into the corresponding element in the array n. The next prompt is for the value which will be written to the output port when an alarm is required; that number is held as the variable out. You are then requested to enter the numbers of the lines on the entry/exit route. "Enter as a sequence:" means that if, for example, you required lines 0,3,5 and 7 to be on the entry/exit route, you would enter 0357. That input string is

PARTS LIST

IC1	– 74LSØØ quad two-input NAND
D1,2	1N4148
R1,2	– 10kΩ ¼W carbon (E12)
R3	– 82kΩ ¼W carbon (E12)
C1	4μ7 63V Elel
C2	– 100nF ceramic disc
IC Socket 14-pin	
Veroboard (105 × 24H)	
Veropins (5)	
Sticky pad, Solder, Wire, etc.	

Figure 1.



then sliced by a FOR-NEXT loop and 1s are written to the elements of the array d which correspond to these lines.

The next loop asks you which input lines should be monitored at all. Pressing ENTER in response to those prompts is the same as pressing "n" followed by ENTER, as the program tests only for a "y". Again, where a line should be monitored, a 1 is written into the corresponding array element of the array m. The program then has all the information it requires to operate.

The following lines poll the keyboard for an "s". When an "s" is detected, the exit delay is started. The delay is controlled by a FOR-NEXT variable l and can be changed to alter the delay time. On each loop a subroutine at line 5000 is called. That routine reads the input port and places the binary value of each line into an element in the array a.

On return, the loop between lines 220 and 240 tests each value. If line 230 detects a fault on a line which is not on the entry/exit route, an alarm will be generated. Once the delay has finished, the program moves into the main polling routine. That is a continuous loop which reads the input port — subroutine 5000 — and tests the lines which should be monitored for a fault condition.

If a fault is detected, the program jumps to line 300 — the entry delay. The entry delay works in exactly the same way as the exit delay, using the same control variable (l). If the program is not stopped during that delay, or if a line outside the entry/exit route is operated, an alarm is generated by lines 6000–6010.

You should have realised that the program contains few refinements, such as error-trapping routines for the data-input stages. That is left to you and we would be interested to hear of any routines you have devised.

The hardware device increases the flexibility of the back-up monitoring. Using the board, you can monitor one NC line and one NO line, as opposed to two NO lines. Further, the device filters those signals to eliminate any transients which could cause a false alarm.

Figure 2.

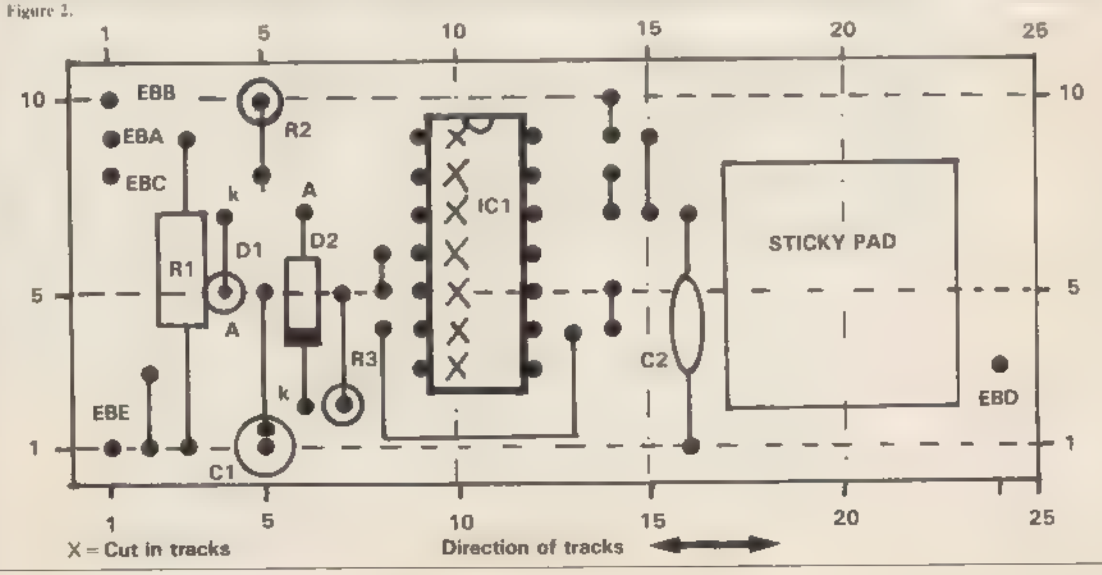
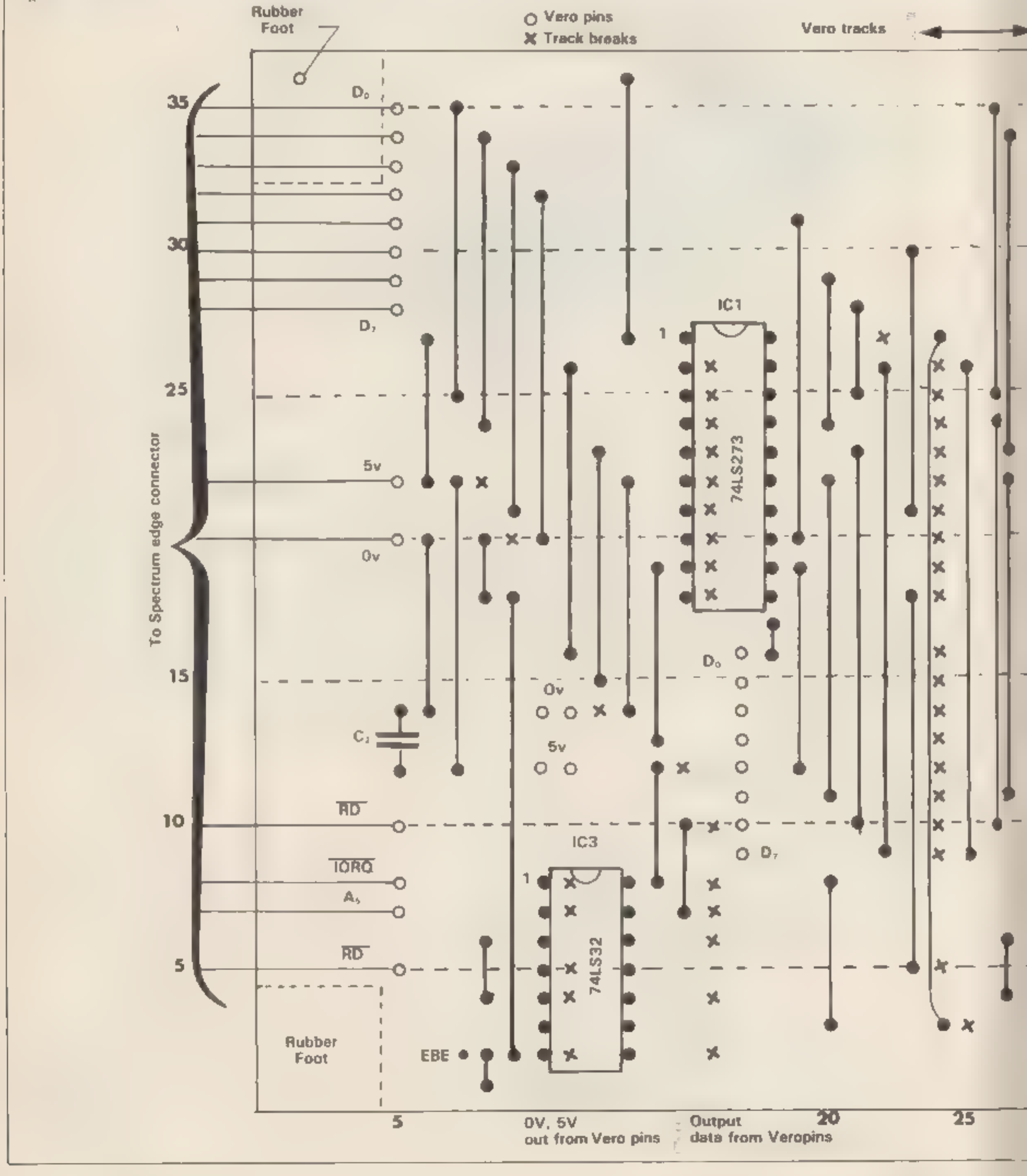
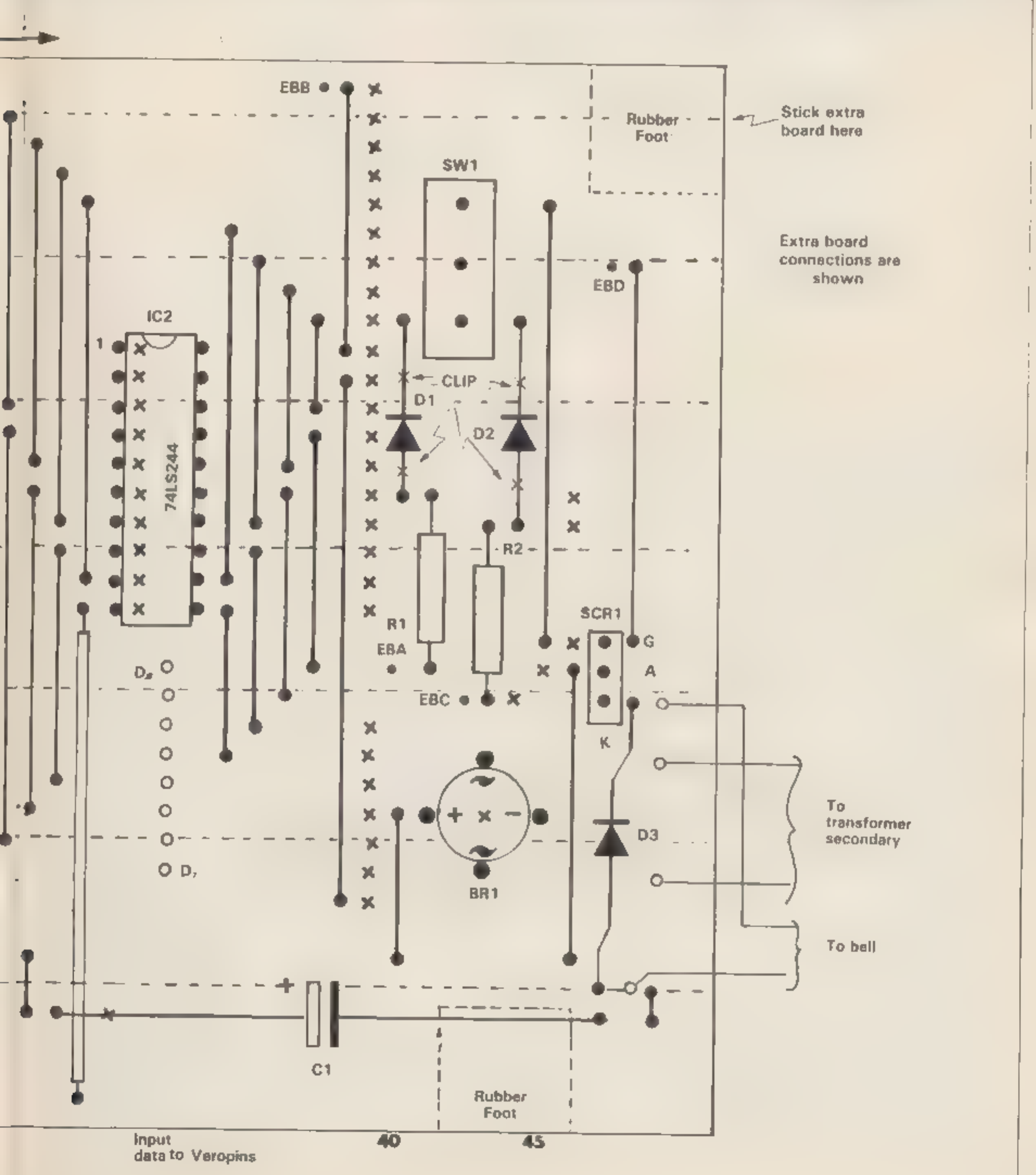


Figure 2.


Rubber Foot
○ Vero pins
✖ Track breaks
Vero tracks
To Spectrum edge connector
35
30
25
15
10
5
D0
D7
5v
0v
IC1
1
74LS273
D0
D7
0v
5v
C1
RD
IORQ
A5
RD
IC3
1
74LS32
EBE
Rubber Foot
5
20
25
0V, 5V out from Vero pins
Output data from Veropins


EBB
Rubber Foot
Stick extra board here
SW1
Extra board connections are shown
EBD
IC2
1
74LS244
CLIP
D1
D2
R2
R1
SCR1
G
A
K
EBA
EBC
D_0
D_7
To transformer secondary
D3
BR1
To bell
C1
Rubber Foot
Input data to Veropins
40
45

BURGLAR ALARM 2

Looking at the circuit — figure one — you can see that the device works around IC1, a 74LS00 quad two-input NAND gate IC. The inputs to gate (a) are both held high, one by a NC switch chain, the other by RZ. In that state, the NAND gate output is low, causing the final output to be low. If any switch is operated, the output of gate (a) goes high, causing C1 to charge via D2 and R3.

If the output of gate (a) remains high for long enough — about 0.5 seconds — the voltage at the input of gate (b) will rise above the switching voltage and the final output will go high, causing an alarm. If, however, gate (a) output goes high for less than 0.5 second, gate (b) will not switch and no alarm will be generated. D1 provides a rapid-discharge path for C1, so that only an incredibly severe series of transients will break through the filter to cause an alarm.

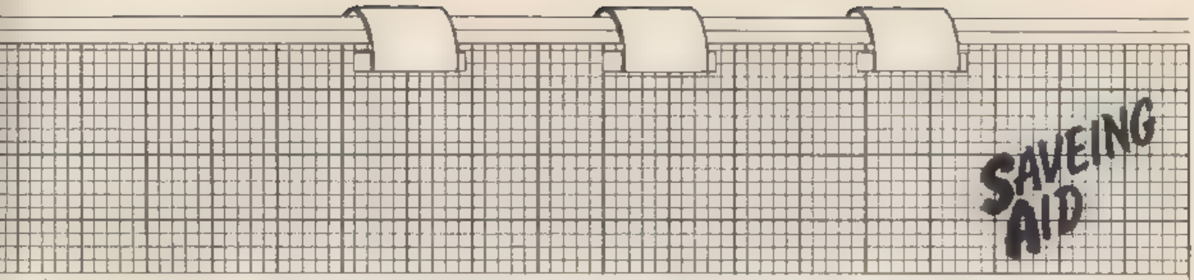
The circuit is built on a small piece of Veroboard — 10S × 24H. A layout diagram is shown in figure two. First solder in the IC socket. Then proceed to solder all discrete components and the five Veropins. Finally, insert IC1 and check your work thoroughly. The original burglar alarm layout diagram is reproduced as figure three with details of where to connect the five flying-leads, labelled EBA to EBE. The functions of those connections are shown in the circuit diagram. Note that there are two breaks to be made in the leads of D1 and D2 on the main board.

The add-on board has been designed so that there is space for a sticky-pad to attach the add-on to the main board. The recommended fixing position is shown in figure three.

Finally, here is the address omitted from the last issue: ADE Security, 217 Warbreck Moor, Aintree, Liverpool, Merseyside L9 0HU. The company will provide a comprehensive catalogue of security devices free if you send a large SAE.

```
  10 REM ALARM CONTROL PROGRAM
  20 DIM m(8): DIM n(8): DIM d(8
)
  30 POKE 23609,255
  40 BORDER 0: PAPER 0: INK 7: C
LS
  50 FOR a=0 TO 7: PRINT "Is lin
e ";a;" no or nc?"
  60 INPUT "Enter 'o' or 'c' ";
LINE a$
  70 IF a$="c" THEN LET n(a+1)=1
  80 PRINT "Line ";a;" is n";a$:
 NEXT a
  90 PAUSE 200: CLS
 100 INPUT "Value at output port
 on alarm ? ";out
 110 PRINT "Which lines on en/ex
 route ?": INPUT "Enter as seque
nce:"; LINE a$
 120 FOR a=1 TO LEN a$: LET d(VA
L a$(a)+1)=1
 130 PRINT "Line ";a$(a);" on en
/ex route.": NEXT a
 140 PAUSE 200: CLS : FOR a=0 TO
 7: PRINT "Monitor line ";a;" ?"
 150 INPUT "Enter y or n: "; LIN
E a$
 160 IF a$="y" THEN LET m(a+1)=1
: PRINT "Line ";a;" will be moni
tered"
 170 NEXT a: PAUSE 200
 180 CLS : PRINT "Press 's' to s
tart ex. delay"
 190 IF INKEY$<>"s" THEN GO TO 1
90
 200 PRINT FLASH 1;"Delay starte
d-leave premises"
 210 FOR l=1 TO 63: GO SUB 5000
 220 FOR q=1 TO 8
 230 IF d(q)=0 AND n(q)<>a(q) TH
EN GO TO 6000
 240 NEXT q: NEXT l
 250 CLS : PRINT "Main moniter l
oop"
 260 GO SUB 5000


 270 FOR q=1 TO 8
 280 IF m(q)=1 AND n(q)<>a(q) TH
EN GO TO 300
 290 NEXT q: GO TO 260
 300 CLS : PRINT "Entry delay in
 operation": FOR l=1 TO 63: GO S
UB 5000
 310 FOR q=1 TO 8: IF d(q)=0 AND
 n(q)<>a(q) THEN GO TO 6000
 320 NEXT q: NEXT l: GO TO 6000
5000 LET n=IN 65503
5010 DIM a(8)
5020 FOR c=1 TO 8
5030 LET a(c)=n-2*INT (n/2)
5040 LET n=INT (n/2)
5050 NEXT c: RETURN
6000 CLS : PRINT FLASH 1;"ALARM
SOUND"
6010 OUT 65503,out: STOP
```



SAVEing a lot of trouble in LOADing

Charles Rowbotham believes many of the difficulties in loading on the ZX-81 are the result of recorders having automatic gain controls.

HAVE YOU a ZX-81 loading problem? After all the advice are you still having difficulties? There is one area which appears to have attracted little attention but which I believe is nevertheless fairly widespread. That belief lies in reading between the lines of readers' letters.

May I suggest that this is really a SAVEing problem and is associated with certain makes of cassette recorder. It was one I encountered some 15 months ago when I bought a new cassette recorder after my old one, which had worked perfectly with both ZX-81 and ZX-80, gave up the ghost.

Nowadays it is the usual practice to manufacture portable cassette recorders featuring automatic gain control for recording, as most people find that is more convenient to use than a manually-adjusted control and it eliminates the necessity of a VU recording meter, thus keeping down the cost of and simplifying manufacture.

There is, however, a price to pay. Nothing is ever perfect and because of that differing types of AGC appear on different machines. There are two important factors to consider, the response or attack time and the recovery or release time of the AGC. The response time is the time the circuit takes to respond to the incoming signal level and adjust it to the correct level for recording. The recovery time is the period needed by the AGC to return the circuit to its full sensitivity.

In the five-second period of silence which precedes the program when SAVEing with the ZX-81, the AGC can regain its full sensitivity; consequently the one millivolt program signal from the ZX-81 may be five times too great and although the response time may be only a few milliseconds, the first fraction of the signal is recorded at far too high a level.

The AGC then reduces the signal level to its optimum level for the remainder of the recording and that level may vary from program to program.

When recording speech or music that does not matter a great deal but for the ZX-81 the consequence can be disastrous. On LOADing the program, the early tape signal may be clipped or distorted, leading to information degradation. The correct part of the tape may produce a signal too low for successful loading. No matter how carefully the volume control is adjusted, the discrepancy between the two signal levels is too great for the ZX-81 for which, according to Sinclair Research, the signal should lie between 1V and 2.5V peak, with 2V peak as the optimum for successful LOADing.

COMPONENTS LIST
One 3.5mm Jack socket
One 3.5mm Jack plug
One 2.2k Linear potentiometer
One Pointer knob
12in. approx. Single-screen cable, 2 or 3mm
One metal box — used tobacco tin or similar
Approx. cost, excluding box, £1.10.

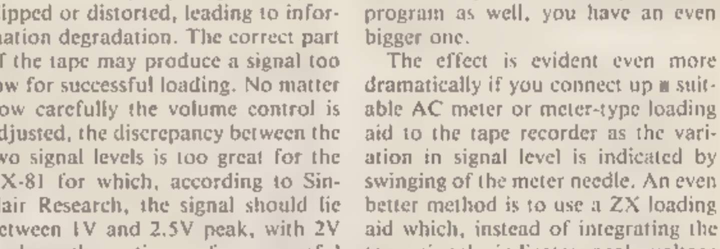
That situation is bad enough but even worse things may happen, as with some recorders the AGC overcompensates and swings too far the opposite way and may even hunt about during the entire recording. Such a tape is almost impossible to load.

To check whether your recordings are faulty, listen carefully to a tape. If the sound level drops after the first fraction of a second you have a problem; if the sound level of the tape alters throughout the main part of the program as well, you have an even bigger one.

The effect is evident even more dramatically if you connect up a suitable AC meter or meter-type loading aid to the tape recorder as the variation in signal level is indicated by swinging of the meter needle. An even better method is to use a ZX loading aid which, instead of integrating the tape signal, indicates peak voltage levels and, because it has no mechanical inertia, gives a much more rapid response.

An irregular flashing of the green and amber light-emitting diodes will be evident, no matter how carefully the volume control is adjusted. Hearing is a logarithmic function, whereas metering is a linear function; a doubling of the signal is only just about

Figure 1. Loading Aid



SAVEING AID

detectable to the human ear. Any signal variation therefore is much more evident when looking at a ZX loading aid or at a meter.

Can anything be done to cure the problem? Take heart, the remedy is simple and inexpensive. All one has to do is to arrange matters so that the AGC is rendered largely inoperative by attenuating the signal from the ZX-81 so that the recording circuit is working at maximum or near maximum sensitivity.

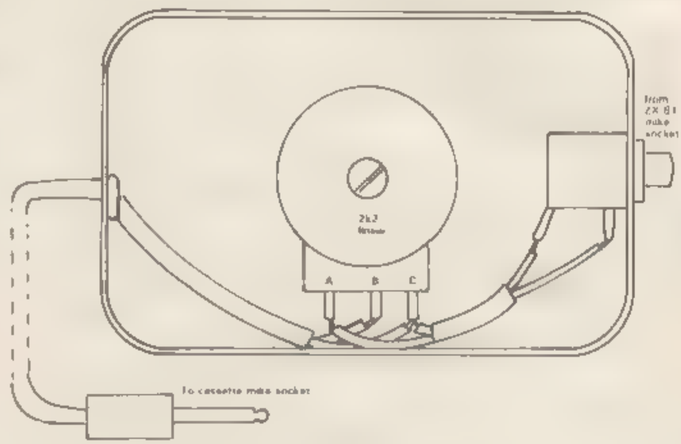
To do that the simple circuit shown in figure one is all that is required. A 2.2K linear potentiometer is connected via a 3.5mm. jack socket and screened lead to the MIC socket of the ZX-81. The centre arm of the potentiometer is connected via a similar lead and a 3.5mm. jack plug to the MIC socket of the cassette recorder. The potentiometer is housed in a metal box — an old tobacco tin is ideal — to screen the unit and avoid picking-up mains hum and other electrical interference.

The method of construction is shown in figure two. Drill a hole in the centre of the base of the tin 0.375in. or 9.5mm. in diameter. In the centre of one end of the box, drill a hole for the jack socket; the hole need be about ¼in. in diameter but its exact size will depend on the type of jack socket you possess.

At the opposite end of the box, drill a hole large enough to pass the screened wire; use of a rubber gromet will prevent the tin cutting into the wire. Fix the potentiometer into the hole in the base of the tin and the jack socket into the ¼in. hole. Cut off about 3in. of screened wire and solder one end of the remaining 9in. to the jack plug, the braid of the wire to the side tag and the centre wire to the centre tag; screw on the plug body and pass the free end of the wire through the hole in the box and tie a knot in it about 2in. from the end.

Solder the braid of that wire and the braid of the short piece of wire to tag C — figure two — of the potentiometer; solder the insulated end of the plug wire to tag B and the centre wire of the short length to tag A. Complete the connections by soldering the free end to the jack socket — braid to the side tag and centre to end tag.

Figure 2. View of inside of box.



If the jack socket is an insulated type, solder a wire from side tag to tin box. Replace the box lid, turn upside down and cut the spindle to the correct length, then fix the knob.

Connect the box between the ZX-81 and your cassette recorder and enter a program into your ZX-81. Turn the potentiometer knob fully anti-clockwise and adjust the volume control of the recorder so that, under normal conditions, it is at a comfortable listening level.

Disconnect the plug from the ear socket of the recorder so that you will be able to hear the program while it is being SAVEd; then SAVE the program in the usual way. While the program is being SAVEd, turn the potentiometer knob slowly in a clockwise direction until you can just hear the program in the cassette speaker; then slowly and carefully continue to turn the knob until the sound from the speaker does not become louder; mark the position of the pointer and repeat the exercise until you are sure of the correct pointer position.

Mark that indelibly, as that is the correct adjustment for the potentiometer for all future recordings of your ZX-81 programs. The AGC circuit is then giving maximum gain and therefore there will be no signal voltage surge after the five-second silent lead-in to the program and the program will record at a steady level throughout, giving a well-modulated tape which can be checked as described.

As an added bonus there will be less noise on the silent lead-in, thus eliminating another cause of LOADing failure.

Should your cassette player be a type which does not allow you to listen while recording but has a VU level meter, use that to find the correct potentiometer setting and turn up the knob slowly until the meter needle goes no higher.

If neither possibility exists you will need to make a graduated scale for the potentiometer knob and make a series of short recordings at different settings close together on the pointer scale, making a note of each setting. Re-wind the tape and listen to it; when the point is reached where the sound is at a maximum, refer to your notes and mark the scale where that occurs.

Should you have difficulty of the type described with your Spectrum, you will find the device equally effective.

SHOPPING LIST

Project buyers' guide

HERE is a list of suppliers for difficult-to-obtain items which have been used in projects

---

74LS133 IC as used in the Latch Card project

MS Components Ltd
Ambit International
Watford Electronics

PCB mounting 3.5mm. jack sockets as used in the Central Heating Controller project.

MS Components Ltd

Edge connectors 23-way for ZX 81 and
28-way for Spectrum

Innovonics

Extender cards for fitting to rear of edge connector to allow stacking add-ons.

23-way for ZX-81	ZXTONGUE
28-way for Spectrum	SPECTONGUE

Innovonics

AY 3-8910 Sound Chip

Cricklewood Electronics
Watford Electronics

Weather Station anemometer
Ribbon cable
DIL headers

Innovonics

MS Components Ltd, Zephyr House, Waring Street, West Norwood, London SE27 Tel: 01-670 4466.

Ambit International, 200 North Service Road, Brentwood, Essex. Tel 0277-230909.

Watford Electronics, 33 34 Cardiff Road, Watford, Herts. Tel 0923-40588.

Innovonics, 147 Upland Road, East Dulwich, London SE22

Cricklewood Electronics Ltd, 40 Cricklewood Broadway London NW2 3ET Tel 01-452 0161.

UPDATE

Errors and mishaps

August/September, page 13 column 2, figures under $\bar{y}$ should be 01234567.

Cassette control, page 16, column 2, fourth line from bottom should read "a line such as xxxx OUT 63, 128: MERGE".

Prowler, page 20. Figure 3 should be figure 2 and labelled battery compartment. Motor supply leads should be labelled "Negative black wire, Positive red wire". Page 21, figure 2, should be figure 3, and labelled "plan view of hull". Page 23, figure 6. Look at figure with Vero tracks running right to left and "front" at left, then add the following. SK3 pin 1 is at bottom right, SK1 pin 1 is at top left, SK2 pin 1 is at top left. The unlabelled arrow by IC1 is "c". D by RP1 should be D_g. The bottom end of RP1 is the indexed end. The two outer links at either side of RL1 should stop one row lower down on a level with the relay pins, and the track break by the left-hand link should also be moved down one row.

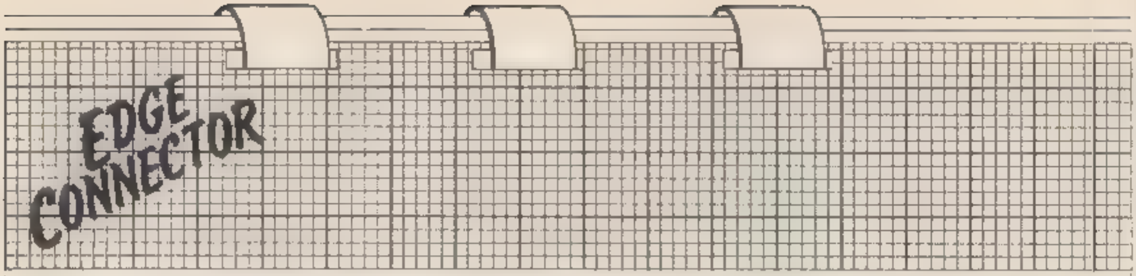
Burglar Alarm, page 28, figure 2. The $\overline{RD}$ connection above $\overline{IORQ}$ should be $\overline{WR}$. Page 30 figure 1. The unlabelled connection between A_5 and $\overline{RD}$ should be $\overline{WR}$.

Real-time clock, page 34, figure 1. On some copies the connections to XTAL1 have disappeared into the fold of the magazine. XTAL1 is connected between CRYS(a) and CRYS(b). Page 39, figure 7. The top left-hand pin by 'a1' is "pin 1". The links to address the board at 65055 were omitted link, pins 2 and 3, pins 5 and 6, pins 8 and 9, pins 11 and 12, pins 14 and 15.

Circuit layout, page 43. Figure 4 resistors should be figure 4a resistors and the other unnumbered figure should be figure 4b capacitors. Page 44, figure 7. The BD939 transistor should be a BD131. Page 45, column 3, paragraph 2, TTC should be TTL. Page 45, figure 8. The 'd' by TR3 should be 'c'. The omitted lead labelling of TR_2 is the same as for TR_1. C_1 is soldered on the underside of the board.

JUNE/JULY

Sound Generator, page 24. Some people have had difficulty making the project work. One reason is the weak signal level of the issue one Spectrum clock. In the next issue of *Sinclair Projects* we will publish some modifications to the board overcoming the problem and also giving access to the AY-3-8910 ports.



Edge Connector signal allocation

SPECTRUM

Side B (Bottom)	Pin	Side A (Top)
A_{14}	1	A_{15}
A_{12}	2	A_{13}
5v	3	D_7
9v	4	
	5	Slot
0v	6	D_0
0v	7	D_1
CK	8	D_2
A_0	9	D_6
A_1	10	D_5
A_2	11	D_3
A_3	12	D_4
$\overline{IORQGE}$	13	$\overline{INT}$
0v	14	$\overline{NMI}$
Video	15	$\overline{HALT}$
Y	16	$\overline{MEMRQ}$
V	17	$\overline{IOREQ}$
U	18	$\overline{RD}$
$\overline{BUSRQ}$	19	$\overline{WR}$
$\overline{RESET}$	20	-5v
A_7	21	$\overline{WAIT}$
A_6	22	+12v
A_5	23	-12v
A_4	24	$\overline{M1}$
$\overline{ROMCS}$	25	$\overline{RFSH}$
$\overline{BUSAK}$	26	A_8
A_9	27	A_{10}
A_{11}	28	

ZX-81

Side B (Bottom)	Pin	Side A (Top)
5v	1	D_7
9v	2	$\overline{RAMCS}$
	3	Slot
0v	4	D_0
0v	5	D_1
Φ	6	D_2
A_0	7	D_6
A_1	8	D_5
A_2	9	D_3
A_3	10	D_4
A_{15}	11	$\overline{INT}$
A_{14}	12	$\overline{NMI}$
A_{13}	13	$\overline{HALT}$
A_{12}	14	$\overline{MREQ}$
A_{11}	15	$\overline{IORQ}$
A_{10}	16	$\overline{RD}$
A_9	17	$\overline{WR}$
A_8	18	$\overline{BUSAK}$
A_7	19	$\overline{WAIT}$
A_6	20	$\overline{BUSRQ}$
A_5	21	$\overline{RESET}$
A_4	22	$\overline{M1}$
$\overline{ROMCS}$	23	$\overline{RFSH}$

PANIC
"It must be a nightmare" he screamed as he ran around the maze of floors, desperately climbing ladders to different levels in a vain attempt to evade the monsters.
He had only his neutron hammer to dig holes for the vile creatures to fall into, but even then he had to repeatedly hit them before they dropped to their deaths. As if that wasn't bad enough, the oxygen supply was running out!
"It's simply impossible" he wailed as the monsters advanced on all sides. .
SPECTRUM 48K
£5·95